

# ۶۵۰ برنامه C# با حل کامل آنها

(حل مسائل C# – مرجع کامل)



مولفین: دکتر رمضان عباس نژادورزی  
مهندس محمد نادعلی زاده چاری

## برخی از عناوین مهم

آشنایی با C# (دستورات ورودی و خروجی)  
ساختار تصمیم و حلقه تکرار  
متدها در C#  
آرایه‌ها و رشته‌ها  
کلاس‌ها و وراثت  
فایل‌های ورودی و خروجی

---

---

# ۶۵۰ برنامه C# با حل آن‌ها

## (حل مسائل C# – مرجع کامل)

---

---

تألیف:

دکتر رمضان عباس نژاد ورزی  
مهندس محمد نادعلی زاده چاری



فن‌آوری نوین

---

---

عنوان و نام پدیدآور	عباس نژاد ورزی، رمضان، ۱۳۴۸ -
مشخصات نشر	۶۵۰ برنامه C# با حل آنها (حل مسایل - C# مرجع کامل) / تألیف رمضان عباس نژاد ورزی، محمد نادعلی زاده چاری.
مشخصات ظاهری	بابل: فناوری نوین، ۱۳۹۷.
شابک	۴۸۸ ص.: مصور، جدول.
وضعیت فهرست نویسی	۴۵۰۰۰۰ ریال ۱-۱۲-۷۲۷۲-۶۰۰-۹۷۸ :
عنوان گسترده	فیپا
موضوع	ششصد و پنجاه برنامه C# با حل آنها (حل مسایل - C# مرجع کامل).
موضوع	سی ++ (زبان برنامه نویسی کامپیوتر)
موضوع	سی ++ (زبان برنامه نویسی کامپیوتر) -- آزمون ها و تمرین ها (عالی)
موضوع	C++ (Computer program language) Exercises, questions, etc. (Higher)
شناسه افزوده	نادعلی زاده چاری، محمد، ۱۳۶۴ -
رده بندی کنگره	۱۳۹۷ ۲۶ع۹۳/س ۷۶/۷۳QA
رده بندی دیویی	۰۰۵
شماره کتابشناسی ملی	۵۱۳۰۴۷۲

تلفن: ۰۱۱-۳۲۲۵۶۶۸۷

[www.fanavarienovin.net](http://www.fanavarienovin.net)

بابل، کد پستی ۴۷۱۶۷-۷۳۴۴۸



### ۶۵۰ برنامه C# با حل آنها (حل مسائل C# - مرجع کامل)

تألیف: رمضان عباس نژاد ورزی - محمد نادعلی زاده

نوبت چاپ: چاپ اول

سال چاپ: بهار ۹۷

شمارگان: ۱۰۰۰

قیمت: ۴۵۰۰۰ تومان

نام چاپخانه و صحافی:

شابک: ۹۷۸-۶۰۰-۷۲۷۲-۱۲-۱

نشانی ناشر: بابل، چهارراه نواب، کاظم بیگی، جنب مسجد منصور کاظم بیگی، طبقه اول

طراح جلد: کانون آگهی و تبلیغات آبان (احمد فرجی)

تهران، خ اردیبهشت، نبش وحید نظری، پلاک ۱۴۲ تلفکس: ۶۶۴۰۰۱۴۴-۶۶۴۰۰۲۲۰

## فهرست مطالب

فصل اول: آشنایی با C# (دستورات ورودی و خروجی) .....	۵
فصل دوم: ساختار تصمیم و حلقه‌های تکرار .....	۳۶
فصل سوم: متدها در C# .....	۱۳۴
فصل چهارم: آرایه‌ها و رشته‌ها .....	۲۳۵
فصل پنجم: کلاس‌ها و وراثت .....	۳۸۳
فصل ششم: فایل‌های ورودی و خروجی .....	۴۳۹
منابع: .....	۴۸۸

نام کتاب	لینک خرید چاپی	لینک خرید الکترونیکی	لینک فایل نمونه
مبانی رایانه و برنامه نویسی به زبان C++	<a href="http://daneshnegar.com/book_380238.html">http://daneshnegar.com/book_380238.html</a>	<a href="http://ktbr.ir/b30588">http://ktbr.ir/b30588</a>	<a href="http://ketabesabz.com/dl/52319">http://ketabesabz.com/dl/52319</a>
آشنایی با مبانی امنیت شبکه (امنیت اطلاعات)	<a href="http://daneshnegar.com/book_371137.html">http://daneshnegar.com/book_371137.html</a>	<a href="http://ktbr.ir/b30327">http://ktbr.ir/b30327</a>	
اصول طراحی پایگاه داده ها	<a href="http://daneshnegar.com/book_371655.html">http://daneshnegar.com/book_371655.html</a>	<a href="http://ktbr.ir/b29943">http://ktbr.ir/b29943</a>	<a href="http://ketabesabz.com/dl/52155">http://ketabesabz.com/dl/52155</a>
آموزش گام به گام برنامه نویسی پایتون	<a href="http://daneshnegar.com/book_392582.html">http://daneshnegar.com/book_392582.html</a>	<a href="http://ktbr.ir/b29984">http://ktbr.ir/b29984</a>	<a href="http://ketabesabz.com/dl/52181">http://ketabesabz.com/dl/52181</a>
آزمایشگاه C++ (حل مسائل کامپیوتر مرجع کامل)	<a href="http://daneshnegar.com/book_392262.html">http://daneshnegar.com/book_392262.html</a>	<a href="http://ktbr.ir/b29982">http://ktbr.ir/b29982</a>	<a href="http://ketabesabz.com/dl/53546">http://ketabesabz.com/dl/53546</a>
C# با LINQ آموزش گام به گام	<a href="http://daneshnegar.com/book_369388.html">http://daneshnegar.com/book_369388.html</a>	<a href="http://ktbr.ir/b28451">http://ktbr.ir/b28451</a>	
C++ ساختمان داده ها با	<a href="http://daneshnegar.com/book_379094.html">http://daneshnegar.com/book_379094.html</a>	<a href="http://ktbr.ir/b29676">http://ktbr.ir/b29676</a>	<a href="http://ketabesabz.com/dl/52156">http://ketabesabz.com/dl/52156</a>
طراحی سیستم های شی گرا با C#	<a href="http://daneshnegar.com/book_374658.html">http://daneshnegar.com/book_374658.html</a>	<a href="http://ktbr.ir/b29621">http://ktbr.ir/b29621</a>	<a href="http://ketabesabz.com/dl/52126">http://ketabesabz.com/dl/52126</a>
مدیریت استراتژیک فناوری اطلاعات	<a href="http://daneshnegar.com/book_374659.html">http://daneshnegar.com/book_374659.html</a>	<a href="http://ktbr.ir/b29779">http://ktbr.ir/b29779</a>	<a href="http://ketabesabz.com/dl/52125">http://ketabesabz.com/dl/52125</a>
گرافیک رایانه ای با زبان C#	<a href="http://daneshnegar.com/book_376021.html">http://daneshnegar.com/book_376021.html</a>	<a href="http://ktbr.ir/b29674">http://ktbr.ir/b29674</a>	<a href="http://ketabesabz.com/dl/51049">http://ketabesabz.com/dl/51049</a>
درس و کنکور پایگاه داده پیشرفته	<a href="http://daneshnegar.com/book_392578.html">http://daneshnegar.com/book_392578.html</a>	<a href="http://ktbr.ir/b29644">http://ktbr.ir/b29644</a>	<a href="http://ketabesabz.com/dl/52102">http://ketabesabz.com/dl/52102</a>
فیزیک الکتروستاتیک	<a href="http://daneshnegar.com/book_379161.html">http://daneshnegar.com/book_379161.html</a>	<a href="http://ktbr.ir/b29680">http://ktbr.ir/b29680</a>	
تجارت الکترونیکی	<a href="http://daneshnegar.com/book_379188.html">http://daneshnegar.com/book_379188.html</a>	<a href="http://ktbr.ir/b29619">http://ktbr.ir/b29619</a>	<a href="http://ketabesabz.com/dl/51013">http://ketabesabz.com/dl/51013</a>
راهنمای کاربردی کاربری برای شبکه های OPNET شبیه سازی کامپیوتر	<a href="http://daneshnegar.com/book_392583.html">http://daneshnegar.com/book_392583.html</a>	<a href="http://ktbr.ir/b28504">http://ktbr.ir/b28504</a>	
درس و کنکور سیستم عامل پیشرفته	<a href="http://daneshnegar.com/book_392580.html">http://daneshnegar.com/book_392580.html</a>	<a href="http://ktbr.ir/b28505">http://ktbr.ir/b28505</a>	<a href="http://ketabesabz.com/dl/52180">http://ketabesabz.com/dl/52180</a>
شبکه های کامپیوتری با رویکرد کاربردی، آزمایشگاه شبیه سازی شبکه	<a href="http://daneshnegar.com/book_392254.html">http://daneshnegar.com/book_392254.html</a>	<a href="http://ktbr.ir/b28528">http://ktbr.ir/b28528</a>	
آزمایشگاه پایگاه SQL Server 2012	<a href="http://daneshnegar.com/book_377301.html">http://daneshnegar.com/book_377301.html</a>	<a href="http://ktbr.ir/b28503">http://ktbr.ir/b28503</a>	<a href="http://ketabesabz.com/dl/53447">http://ketabesabz.com/dl/53447</a>

<a href="http://ketabesabz.com/dl/۵۳۴۴۶">http://ketabesabz.com/dl/۵۳۴۴۶</a>	<a href="http://ktbr.ir/b۲۸۴۵۰">http://ktbr.ir/b۲۸۴۵۰</a>	<a href="http://daneshnegar.com/book_۳۷۵۸۹۲.html">http://daneshnegar.com/book_۳۷۵۸۹۲.html</a>	کاربرد رایانه در مدیریت و حسابداری
<a href="http://ketabesabz.com/dl/۵۳۴۸۸">http://ketabesabz.com/dl/۵۳۴۸۸</a>	<a href="http://ktbr.ir/b۲۸۴۴۹">http://ktbr.ir/b۲۸۴۴۹</a>	<a href="http://daneshnegar.com/book_۳۶۸۹۲۹.html">http://daneshnegar.com/book_۳۶۸۹۲۹.html</a>	آموزش گام به گام برنامه نویسی بانک اطلاعاتی با ویژوال بیسکانت
	<a href="http://ktbr.ir/b۲۸۴۵۰۲">http://ktbr.ir/b۲۸۴۵۰۲</a>	<a href="http://daneshnegar.com/book_۳۸۰۲۳۸.html">http://daneshnegar.com/book_۳۸۰۲۳۸.html</a>	آموزش گام به گام برنامه نویسی به زبان C++
<a href="http://ketabesabz.com/dl/۵۱۰۴۷">http://ketabesabz.com/dl/۵۱۰۴۷</a>	<a href="http://ktbr.ir/b۲۸۴۴۸">http://ktbr.ir/b۲۸۴۴۸</a>	<a href="http://daneshnegar.com/book_۳۶۸۴۸۶.html">http://daneshnegar.com/book_۳۶۸۴۸۶.html</a>	دانلود کتاب آموزش گام به گام برنامه نویسی بانک اطلاعاتی با C#
	<a href="http://ktbr.ir/b۲۸۳۹۸">http://ktbr.ir/b۲۸۳۹۸</a>		حل مسائل پاسکال
<a href="http://ketabesabz.com/dl/۵۱۰۴۸">http://ketabesabz.com/dl/۵۱۰۴۸</a>	<a href="http://ktbr.ir/b۲۸۴۴۰۱">http://ktbr.ir/b۲۸۴۴۰۱</a>	<a href="http://daneshnegar.com/book_۳۹۲۲۶۲.html">http://daneshnegar.com/book_۳۹۲۲۶۲.html</a>	حل مسائل C++
<a href="http://ketabesabz.com/dl/۵۱۰۱۱">http://ketabesabz.com/dl/۵۱۰۱۱</a>	<a href="http://ktbr.ir/b۲۸۳۹۹">http://ktbr.ir/b۲۸۳۹۹</a>	<a href="http://daneshnegar.com/book_۳۷۴۶۵۷.html">http://daneshnegar.com/book_۳۷۴۶۵۷.html</a>	دانلود کتاب حل مسائل C#
	<a href="http://ktbr.ir/b۳۲۶۷۲">http://ktbr.ir/b۳۲۶۷۲</a>	<a href="http://daneshnegar.com/book_۳۹۲۸۱۸.html">http://daneshnegar.com/book_۳۹۲۸۱۸.html</a>	یا حل کامل C++ برنامه ۶۰۰ مرجع - C++ آنها(حل مسائل) کامل

۶۰۰ برنامه C# با حل آن ها ۷

	<a href="http://ktbr.ir/b32671">http://ktbr.ir/b32671</a>	<a href="http://daneshnegar.com/book392817.html">http://daneshnegar.com/book392817.html</a>	حل مسائل جاوا (حل ۶۰۰ برنامه - مرجع کامل)
		<a href="http://daneshnegar.com/book392815.html">http://daneshnegar.com/book392815.html</a>	۶۵۰ برنامه # C با حل کامل آنها (حل مسائل #-C مرجع).
		<a href="http://daneshnegar.com/book392819.html">http://daneshnegar.com/book392819.html</a>	حل مسائل پایتون (حل ۶۵۰ برنامه )-مرجع کامل
		<a href="http://daneshnegar.com/book392814.html">http://daneshnegar.com/book392814.html</a>	دانش ثروتمند شدن
		<a href="http://daneshnegar.com/book392816.html">http://daneshnegar.com/book392816.html</a>	برنامه سازی پیشرفته به زبان C#
		<a href="http://daneshnegar.com/book392813.html">http://daneshnegar.com/book392813.html</a>	الگوریتم ها و محاسبات موازی
	<a href="http://ktbr.ir/b28397">http://ktbr.ir/b28397</a>		دانلود کتاب حل مسائل C

## مقدمه

زبان C# (سی شارپ) یکی از زبان‌های شی گرا است که در سال‌های اخیر طرفداران زیادی پیدا کرده است. با زبان C# می‌توان برنامه‌های سیستمی از قبیل کامپایلر، مدیریت شبکه، بانک‌های اطلاعاتی و برنامه اندرویدی نوشت. به همین دلیل زبان C# به‌عنوان یکی از زبان‌های مهم در درس برنامه‌سازی پیشرفته در رشته‌های کامپیوتر و فناوری اطلاعات تدریس می‌شود.

این کتاب شامل ۶ فصل است که مفاهیم پایه‌ای زبان برنامه‌سازی C# را تحت کنسول آموزش می‌دهد. فصل‌های این کتاب عبارت‌اند از:

فصل اول، مباحثی نظیر آشنایی با C#، عملگرها و دستورات ورودی خروجی را با ۷۵ مثال برنامه‌نویسی و حل آن‌ها آموزش داده است.

فصل دوم، مباحثی مانند ساختار تصمیم (دستورات if, switch) و ساختار تکرار (دستورات for, while, do while, foreach, continue و break) را با حدود ۲۳۵ مثال کاربردی برنامه‌نویسی آموزش داده است.

فصل سوم، تعریف متدها، فراخوانی متدها و متدهای کتابخانه‌ای را با ۱۰۷ مثال کاربردی شرح داده است.

فصل چهارم، آرایه‌های یک و دوبعدی، رشته‌ها را با ۱۳۵ مثال کاربردی توضیح داده است.

فصل پنجم، کلاس‌ها، وراثت، چندریختی و تعریف مجدد عملگرها را با ۳۹ مثال آموزش داده است.

فصل ششم، فایل‌های ورودی و خروجی متنی و باینری را با ۴۹ مثال به‌صورت پایه‌ای شرح داده است.

این کتاب برای دانشجویانی که می‌خواهند مباحث پایه‌ای C# را یاد بگیرند. مفید است و کتاب طراحی سیستم‌های شی گرا به زبان C# را کامل می‌کند (تمام مسائل آن کتاب را نیز حل کرده است).

در ضمن جهت کاهش حجم کتاب برخی از مسائل و حل آن‌ها به صورت الکترونیکی ارائه شده است. از تمامی اساتید و دانشجویان عزیز تقاضا داریم، هرگونه اشکال، ابهام در متن کتاب، پیشنهاد و انتقادات را به آدرس پست الکترونیک [fanavarienovin@gmail.com](mailto:fanavarienovin@gmail.com) ارسال نمایند.

در پایان امیدوارم این اثر مورد توجه جامعه انفورماتیک کشور، اساتید و دانشجویان عزیز قرار گیرد.

مؤلفین

[fanavarienovin@gmail.com](mailto:fanavarienovin@gmail.com)

## آشنایی با C# (دستورات ورودی و خروجی)

۱. دستوراتی که متغیرهای  $m$  و  $n$  را از نوع `int`،  $ch1$  و  $ch2$  را از نوع `char`،  $d1$  و  $d2$  را از نوع `double` و ثابت  $pl$  را از نوع `float` با مقدار ۳,۱۴ تعریف می‌کند.

۲. دستورات زیر چند بایت از حافظه را اشغال می‌کنند (در محیط‌های ۱۶ بیتی)؟

```
long double x, y;
char ch;
float f1, f2;
```

این دستورات، ۲۹ بایت از حافظه را اشغال می‌کنند. زیرا هر یک از متغیرهای  $x$  و  $y$ ، ۱۰ بایت (نوع `long double`)،  $ch$  یک بایت (نوع `char`) و  $f1$  و  $f2$  هر یک ۴ بایت (نوع `float`) را اشغال می‌کنند. لذا، این دستورات مجموعاً ۲۹ ( $20 + 1 + 8$ ) بایت را اشغال می‌کنند.

۳. عبارت زیر را به ازای  $x = 5$  و  $y = 4$  ارزیابی کنید:

```
m = x + (y * 2) / 6 + y / 8;
```

مقادیر را در عبارت قرار می‌دهیم:

$$m = 5 + (4 * 2) / 6 + 4 / 8$$

$$m = 5 + 8 / 6 + 4 / 8$$

$$m = 5 + 1 + 4 / 8$$

$$m = 5 + 1 + 0 = 6$$

۴. عبارت زیر را به ازای  $x = 4$  و  $y = 3$  ارزیابی کنید:

```
m = x > y ? x * 3 : y * 3;
```

ابتدا مقادیر را در  $x$  و  $y$  قرار می‌دهیم:

$$m = 4 > 3 ? 3 * 3 : 4 * 3;$$

عبارت  $4 > 3$  ارزیابی می‌شود. چون این شرط ارزش درستی دارد،  $3 * 3$  (۹) در  $m$  قرار می‌گیرد.

۵. پس از اجرای دستورات زیر، چه مقداری در  $x$  قرار می‌گیرد؟

```
int x;
long double y;
x = sizeof y;
```

دستور اول، متغیر  $x$  را از نوع `int` تعریف می‌کند، دستور دوم، متغیر  $y$  را از نوع `long double` تعریف می‌کند و دستور سوم، طول نوع متغیر  $y$  را محاسبه کرده، در متغیر  $x$  قرار می‌دهد. بنابراین، ۱۰ در  $x$  قرار می‌گیرد (نوع `long double`، ۱۰ بایت را اشغال می‌کند).

۶. عبارت زیر را بدون استفاده از عملگر `*` انجام دهید:

$$y = 255 * x - 18 * x + 15;$$

ابتدا  $x * 255$  را محاسبه می‌کنیم. برای انجام این کار،  $x$  را ۸ بار به سمت چپ شیفت داده، از  $x$  کم می‌کنیم. زیرا، با هر بار شیفت، مقدار  $x$ ، در ۲ ضرب می‌شود. پس مقدار  $x$  در  $2^8$  (۲۵۶) ضرب می‌گردد. سپس، مقدار  $x * 18 -$  را محاسبه کرده، ( $x$  را ۴ بار به سمت چپ شیفت داده دو بار با  $x$  جمع کرده، علامت آن را به منفی تبدیل می‌کنیم) در پایان، این مقادیر را با ۱۵ جمع می‌کنیم. یعنی، عبارت زیر را داریم:

$y = x \ll 8;$

$y1 = y - x;$

$y2 = x \ll 4;$

$y2 += (x + x);$

$y = y1 - y2 + 15;$

۷. پس از اجرای عبارت زیر مقدار متغیر  $m$  چه خواهد شد؟

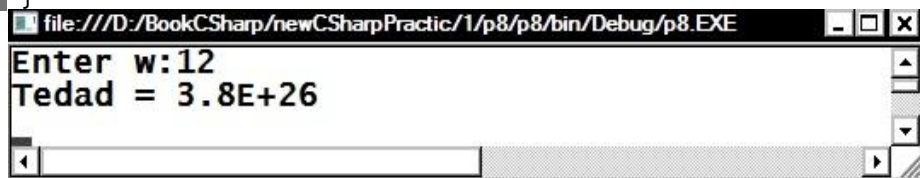
$m = (y = 0, x = ++y, m = y, y + 2);$

ابتدا  $y$  برابر با صفر می‌شود و در عبارت دوم به  $y$  یک واحد اضافه‌شده، در  $x$  قرار می‌گیرد. پس اکنون مقدار  $x$  و  $y$  برابر یک است. در عبارت بعدی،  $y$  در  $m$  قرار می‌گیرد و در نتیجه،  $m$  نیز برابر با یک می‌شود. حاصل عبارت  $y + 2$  که برابر با ۳ است در  $m$  قرار می‌گیرد.

۸. وزن یک مولکول آب  $3.0 \times 10^{-23}$  گرم و وزن یک لیتر آب در حدود ۹۵۰ گرم است. برنامه‌ای که وزن آب را برحسب لیتر از ورودی خوانده، تعداد مولکول‌های آن را محاسبه می‌کند.

```
using System;
namespace p8
{
    class Program
    {
        static void Main(string[] args)
        {
            double m=3.0e-23,l=950,tedad;
            long w;
            Console.Write("Enter w:");
            w = long.Parse(Console.ReadLine());
            tedad=(w*l)/m;
            Console.WriteLine("Tedad = {0}", tedad);
            Console.ReadKey();
        }
    }
}
```

متغیر	هدف
m	وزن یک مولکول
l	وزن یک لیتر آب
w	وزن آب به لیتر از رودی خوانده می‌شود.
tedad	تعداد مولکول‌های آب

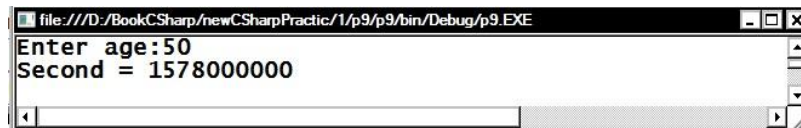


۹. هرسال برابر با  $3/156 \times 10^7$  ثانیه است. برنامه‌ای که سن شما را دریافت کرده، به ثانیه تبدیل می‌کند.

## ۶۰۰ برنامه C# با حل آن‌ها ۱۱

```
using System;
namespace p9
{
    class Program
    {
        static void Main(string[] args)
        {
            double ss = 3.156e7, second;
            int age;
            Console.Write( "Enter age:");
            age = Convert.ToInt32(Console.ReadLine());
            second = age * ss;
            Console.WriteLine( "Second = {0}" , second);
            Console.ReadKey();
        }
    }
}
```

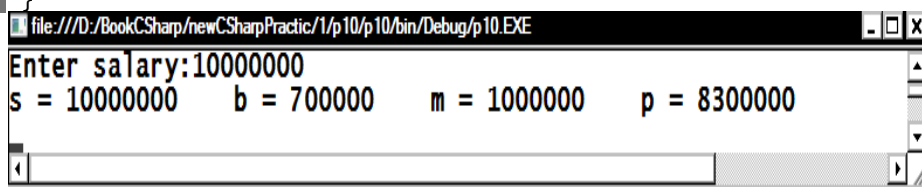
متغیر	هدف
ss	هرسال برحسب ثانیه
age	سن به سال
second	سن برحسب ثانیه



۱۰. برنامه‌ای که حقوق کارمندی را خوانده، بیمه، مالیات و دریافتی او را محاسبه کرده، نمایش می‌دهد. بیمه و مالیات به ترتیب ۷ و ۱۰ درصد حقوق می‌باشند.

```
using System;
namespace p10
{
    class Program
    {
        static void Main(string[] args)
        {
            long p,s,m,b;
            Console.Write("Enter salary:");
            s = long.Parse(Console.ReadLine());
            b=s*7 /100;
            m=s/10;
            p=s-b-m;
            Console.WriteLine("s = {0} b = {1} m = {2} p = {3}"
                , s, b, m, p);
            Console.ReadKey();
        }
    }
}
```

متغیر	هدف
s	حقوق
b	بیمه
m	مالیات
p	دریافتی



۱۱. برنامه‌ای که قیمت کالایی را در سال قبل و جاری خوانده، نرخ تورم و قیمت سال بعد آن را چاپ می‌کند. نرخ تورم باید به صورت درصد (مثلاً ۶/۵٪) حساب گردد. (توضیح: برای محاسبه نرخ تورم و قیمت سال بعد به صورت زیر عمل کنید):

$$\text{نرخ تورم} = \frac{\text{قیمت سال قبل} - \text{قیمت سال جاری}}{\text{قیمت سال قبل}}$$

$$\text{نرخ تورم} \times \text{قیمت سال جاری} + \text{قیمت سال قبل} = \text{قیمت سال بعد}$$

```
using System;
namespace p11
{
    class Program
    {
        static void Main(string[] args)
        {
            double t, y1, y2, y3;
            Console.Write("Enter price for year1:");
            y1 = double.Parse(Console.ReadLine());
            Console.Write("Enter price for year2:");
            y2 = double.Parse(Console.ReadLine());
            t = (float)(y2-y1)/y1;
            y3 = y2 + y2 * t;
            Console.WriteLine("Extera = {0}\t\t Price next year = {1}", t, y3);
            Console.ReadKey();
        }
    }
}
```

متغیر	هدف
t	نرخ تورم
y1	قیمت کالا در سال قبل
y2	قیمت کالا در سال جاری
y3	قیمت تخمینی کالا در سال بعد

۱. برنامه‌ای که شماره دانشجویی و معدل تعداد N دانشجو را از ورودی خوانده، دانشجویی که دومین معدل را از نظر بزرگی دارد، پیدا می‌کند و به خروجی می‌برد. (توضیح: چون هیچ یک از دانشجویان معدل شان کوچک تر از صفر نمی‌باشد، لذا اولین بار معدل دو دانشجویی که بالاترین معدل را دارند، صفر در نظر گرفتیم.)

```
using System;
namespace p1
{
    class Program
    {
        static void Main(string[] args)
        {
            int id1=-1, id2=0, i, n, id;
            float aver, max1=-1, max2=0;
            Console.Write("Enter n:");
            n = Convert.ToInt32(Console.ReadLine());
            if (n < 2)
```

متغیر	هدف
n	تعداد دانشجویان
max1	اولین معدل
max2	دومین معدل
i	شمارنده
id1	شماره دانشجویی که اولین معدل را دارد.
id2	شماره دانشجویی که دومین معدل را دارد.
aver	معدل دانشجو
id	شماره دانشجو

## ۶۰۰ برنامه C# با حل آن‌ها ۱۳

```
{
    Console.WriteLine("Please enter a number greater than 1");
    Console.ReadKey();
    return ;
}
for (i=1; i<= n; i++)
{
    Console.Write( "Enter a id:");
    id = int.Parse(Console.ReadLine());
    Console.Write( "Enter a average:");
    aver = float.Parse(Console.ReadLine());
    if (aver>max1)
    {
        id2=id1;
        max2 = max1;
        max1 = aver;
        id1 = id;
    }
    else if(aver > max2)
    {
        max2=aver;
        id2=id;
    }
}
Console.WriteLine("Max2 = {0}\tId2 = {1}", max2, id2);
Console.ReadKey();
}
}
```

```
file:///D:/BookCSharp/newCSharpPractic/2/p1/p1/bin/Debug/p1.EXE
Enter n:5
Enter a id:1
Enter a average:14.01
Enter a id:2
Enter a average:15.75
Enter a id:3
Enter a average:8.09
Enter a id:4
Enter a average:17.01
Enter a id:5
Enter a average:15.25
Max2 = 15.75      Id2 = 2
```

۲. برنامه‌ای که اعدادی را از ورودی خوانده تشخیص می‌دهد که آیا اعداد موردنظر کامل هستند یا خیر. عددی کامل است که مجموع مقسوم‌علیه‌های آن (به‌جز خودش) برابر با آن عدد باشد، پس از بررسی هر عدد از کاربر سؤال کند که می‌خواهد به کارش ادامه دهد یا خیر.

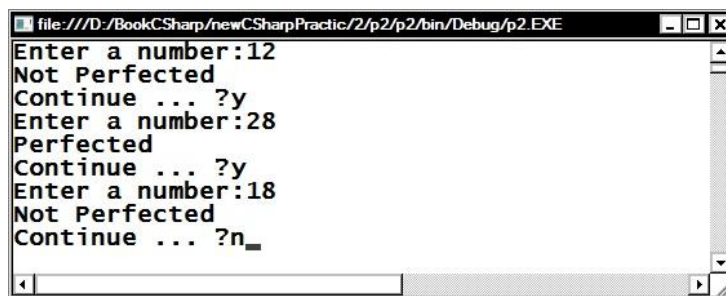
```
using System;
namespace p2
{
    class Program
    {
        static void Main(string[] args)
        {
            int sum, num, i;
            char yes;
            do
            {
                Console.Write ("Enter a number:");
                num = Convert.ToInt32(Console.ReadLine());
                sum = 0;
```

متغیر	هدف
num	اعداد موردنظر
i	شمارنده از ۱ تا num
sum	مجموع مقسوم‌علیه‌ها
yes	تعیین ادامه کار

```

        for (i = 1; i < num; i++)
            if (num % i == 0) sum += i;
        if (sum == num)
            Console.WriteLine( "Perfected");
        else
            Console.WriteLine("Not Perfected");
        Console.Write("Continue ... ?");
        yes =char.Parse(Console.ReadLine());
        yes = char.ToLower(yes);
    }
    while (yes == 'y');
}
}
}

```



۳. برنامه‌ای که تعداد n جمله از سری فیبوناچی را تولید می‌کند.

۱ ۲ ۳ ۵ ۸ ۱۳ ... = سری فیبوناچی

متغیر	هدف
I	شمارنده
f1	جملات سری
f2	جملات سری
f3	جملات سری
N	تعداد جملات

```

using System;
namespace p3
{
    class Program
    {
        static void Main(string[] args)
        {
            int n, f1 = 1, f2 = 1, f3, i;
            Console.Write("Enter a number:");
            n = Convert.ToInt32(Console.ReadLine());
            if (n == 1)
            {
                Console.Write(f1);
                return ;
            }
            else if (n == 2)
            {
                Console.Write("{0} {1} ", f1, f2 );
                return;
            }
            Console.Write("{0} {1} ", f1, f2);
            for (i = 3; i <= n; i++)
            {
                f3 = f1 + f2;
                Console.Write("{0} ", f3);
                f1 = f2;
                f2 = f3;
            }
        }
    }
}

```

## ۶۰۰ برنامه C# با حل آن‌ها ۱۵

```

    }
    Console.ReadKey();
}
}
}

```

۱. برنامه‌ای که شماره کارمندی و حقوق تعدادی از کارکنان موسسه را دریافت کرده، بر اساس تعریف زیر، مالیات حقوق را محاسبه می‌کند و به خروجی می‌برد. سپس مشخص می‌کند بیش‌ترین دریافتی مربوط به کدام کارمند است.

مالیات از معاف ۴۰۰۰۰۰ ≤ حقوق

۱۰ درصد به مازاد بر ۴۰۰۰۰۰ ۵۰۰۰۰۰ ≤ حقوق < ۴۰۰۰۰۰

۱۵ درصد مازاد ۵۰۰۰۰۰ ۷۰۰۰۰۰ ≤ حقوق < ۵۰۰۰۰۰

۱۷ درصد مازاد ۷۰۰۰۰۰ > حقوق

```

using System;
namespace p4
{
    class Program
    {
        static void Main(string[] args)
        {
            long i,salary,tax=0,pay,max=0,
            id1=0, id=-1, n;
            Console.Write("Enter a
            number:");
            n = Convert.ToInt32
            (Console.ReadLine());
            for (i = 1; i <= n; i++)
            {
                Console.Write("Enter a id:");
                id = long.Parse(Console.ReadLine());
                Console.Write("Enter a salary:");
                salary = long.Parse(Console.ReadLine());
                if (salary <= 400000)
                    tax=0;
                else if (salary>400001 && salary<=500000)
                    tax=(salary-400000)* 10 / 100;
                else if(salary>=500001 && salary<=700000)
                    tax=(500000-400001)*10/100+(salary-500001)*15/100;
                else if(salary>=700001)
                    tax=(500000-400001)*10/100+
                    (salary-500001)*15/100+(salary-700001)*17/100;
                pay=salary-tax;
                if (pay > max)
                {
                    max=pay;
                    id1=id;
                }
                Console.WriteLine("Salary={0}Tax={1}Pay=
                {2}", salary, tax, pay);
            }
            Console.WriteLine("Max={0} Id = {1}", max, id1);
            Console.ReadKey();
        }
    }
}

```

متغیر	هدف
Id	شماره کارمندی
Salary	حقوق
Tax	مالیات
Pay	دریافتی
I	شمارنده
id\	شماره کارمندی که بیشترین دریافتی را دارد
Max	بیشترین دریافتی
N	تعداد کارمندان

۱. برنامه‌ای که  $n$  جمله سری فیبوناچی را به روش بازگشتی تولید می‌کند.

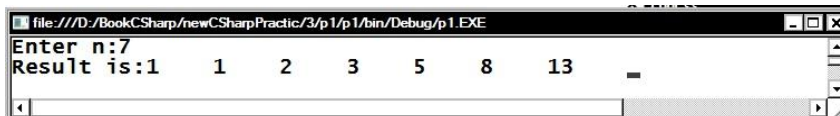
وظایف متدها:

متد **main**: این متد متغیرهای برنامه را تعریف کرده، متد **fibonacci** را فراخوانی می‌کند.

متد **fibonacci**: این متد  $i$ امین عدد فیبوناچی را تولید می‌کند.  $i$  را به عنوان پارامتر می‌پذیرد.

```
using System;
namespace p1
{
    class Program
    {
        static int fibo(int n)
        {
            if (n == 1)
                return (1);
            if (n == 2)
                return (1);
            return (fibo(n - 2) + fibo(n - 1));
        }
        static void Main(string[] args)
        {
            int n, i;
            Console.WriteLine("Enter n:");
            n = int.Parse(Console.ReadLine());
            Console.WriteLine("Result is:");
            for (i = 1; i <= n; i++)
                Console.WriteLine("{0} ", fibo(i));
            Console.ReadKey();
        }
    }
}
```

متد	متغیر	هدف
Main	n	تعداد اعداد فیبوناچی
	i	شمارنده حلقه



۲. برنامه‌ای که فاصله‌ای را برحسب فوت و اینچ دریافت کرده، معادل آن را برحسب متر و سانتی‌متر بیان می‌کند. هر فوت ۰/۳۰۴۸ متر، یک متر ۱۰۰ سانتی‌متر و هر فوت ۱۲ اینچ است. حداقل از سه متد استفاده کنید. یکی برای ورودی، یکی برای انجام محاسبات و یکی برای خروجی.

وظایف متدها:

متد **main**: این متد، متدها **read**، **write** و **calculate** را فراخوانی می‌کند.

متد **read**: این متد فاصله را به فوت و اینچ می‌خواند.

متد **calculate**: این متد، هر یک از فواصل برحسب فوت و اینچ را به متر و سانتی‌متر تبدیل می‌کند.

متد **write**: این متد برای نمایش فواصل تبدیل‌شده به کار می‌رود.

متد	متغیر	هدف
عمومی	foot	ورودی به فوت
	Inch	ورودی به اینچ
	foot_to_meter	فوت به متر
	inch_to_meter	اینچ به متر
	centi_meter	سانتیمتر
	foot_to_centimeter	فوت به سانتیمتر

```
using System;
namespace p2
{
    class Program
    {
        static int foot;
        static int inch;
        static float foot_to_meter;
        static float foot_to_centimeter;
        static float inch_to_meter;
        static float inch_to_centimeter;

        static void read()
        {
            Console.Write("Foot?");
            foot = int.Parse(Console.ReadLine());
            Console.Write("Inch?");
            inch = int.Parse(Console.ReadLine());
        }

        static void calculate()
        {
            foot_to_meter=0.3048f*foot;
            foot_to_centimeter=100*foot_to_meter;
            inch_to_meter=(float) (1.0f/12)*0.3048f*inch;
            inch_to_centimeter=100*inch_to_meter;
        }

        static void write()
        {
            Console.WriteLine("The {0} foot is {1} meter", foot,
                foot_to_meter);
            Console.WriteLine("The {0} foot is {1} centiMeter", foot,
                foot_to_centimeter);
            Console.WriteLine("The {0} inch is {1} meter", inch,
                inch_to_meter);
            Console.WriteLine("The {0} inch is {0} centiMeter", inch,
                inch_to_centimeter);
        }

        static void Main(string[] args)
        {
            read();
            calculate();
            write();
            Console.ReadKey();
        }
    }
}
```

```
file:///D:/BookCSharp/newCSharpPractic/3/p2/p2/bin/Debug/p2.EXE
Foot?100
Inch?200
The 100 foot is 30.48 meter
The 100 foot is 3048 centiMeter
The 200 inch is 5.08 meter
The 200 inch is 200 centiMeter
```

۳. نیروی جاذبه بین دو جسم به صورت زیر محاسبه می گردد:

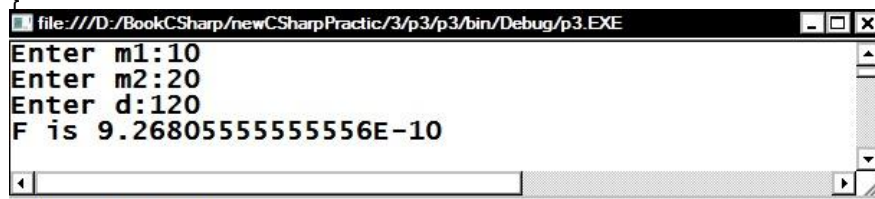
$$f = \frac{g * m1 * m2}{d^2}$$

$m_1$  جرم جسم اول،  $m_2$  جرم جسم دوم،  $d$  فاصله بین دو جسم و  $g$  ثابت جهانی جاذبه با مقدار  $6.673 \times 10^{-8}$  cm(g,sec) است. برنامه ای که با استفاده از دو متد مقدار نیروی جاذبه محاسبه می کند.

متد	متغیر	هدف
سراسری	g	نیروی جاذبه جهانی
main	m1	جرم جسم اول
	m2	جرم جسم دوم
	D	فاصله بین دو جسم

```
using System;
namespace p3
{
    class Program
    {
        const double g = 6.673e-8;
        static double calc_f(double m1, double m2, double d)
        {
            return (g * m1 * m2 / pow(d));
        }
        static double pow(double d)
        {
            return (d * d);
        }

        static void Main(string[] args)
        {
            double m1,m2,d;
            Console.Write("Enter m1:");
            m1=double.Parse(Console.ReadLine());
            Console.Write("Enter m2:");
            m2=double.Parse(Console.ReadLine());
            Console.Write("Enter d:");
            d=double.Parse(Console.ReadLine());
            Console.WriteLine("F is {0}", calc_f(m1,m2,d));
            Console.ReadKey();
        }
    }
}
```



۴. یکی از مشکلات هواشناسی محاسبه شاخص سردی باد است. برای این منظور، فرمول زیر وجود دارد:

$$w = 33 - \frac{(10\sqrt{V} - V + 10.5)(33 - t)}{23 - 1}$$

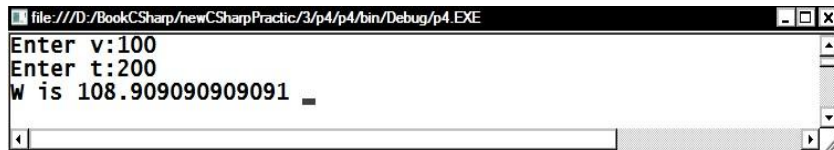
$V$ ، سرعت باید برحسب متر بر ثانیه،  $t$  دما برحسب درجه سانتی گراد، و  $(t \leq 10)$  و  $W$  شاخص سردی باد است. برنامه ای که با استفاده از یک متد، سرعت باد و دما را می خواند، شاخص سردی باد را نمایش می دهد.

متد	متغیر	هدف
Main	v	سرعت باد

## ۶۰۰ برنامه C# با حل آن‌ها ۱۹

```
using System;
namespace p4
{
    class Program
    {
        static double calc_cold(double v, double t)
        {
            return (33 - ((10 * Math.Sqrt(v) - v + 10.5) *
                (33 - t)) / 23.1);
        }
        static void Main(string[] args)
        {
            double v,t;
            Console.Write("Enter v:");
            v = double.Parse(Console.ReadLine());
            Console.Write("Enter t:");
            t = double.Parse(Console.ReadLine());
            Console.Write("W is {0} ", calc_cold(v,t));
            Console.ReadKey();
        }
    }
}
```

دما	T	
-----	---	--

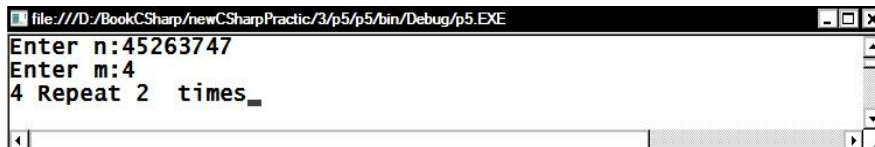


۵. برنامه‌ای که یک عدد و یک رقم را خوانده، با استفاده از متدی تعداد تکرار رقم در عدد را برمی‌گرداند و چاپ می‌کند.

```
using System;
namespace p5
{
    class Program
    {
        static int repeat(long n,int m)
        {
            int count=0;
            while (n>0)
            {
                if ((n % 10 ) == m) count++;
                n/=10;
            }
            return(count);
        }
        static void Main(string[] args)
        {
            long n;
            int m;
            Console.Write("Enter n:");
            n = long.Parse(Console.ReadLine());
            Console.Write("Enter m:");
            m = int.Parse(Console.ReadLine());
            Console.Write("{0} Repeat {1} times", m, repeat(n, m));
            Console.ReadKey();
        }
    }
}
```

هدف	متغیر	متد
عدد موردنظر	n	main
رقمی که باید در عدد جستجو گردد	m	
تعداد تکرار رقم در عدد	count	repeat

```
    }
  }
}
```



۶. برنامه‌ای که  $x$  و  $n$  را خوانده، مقدار سینوس زاویه  $x$  را محاسبه می‌کند.

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots + \frac{x^n}{n!}$$

وظایف متدها:

متد **main**: این متد  $x$  و  $n$  را خوانده، با فراخوانی متدها **fact** و **pow** حاصل عبارت  $\sin(x)$  را چاپ می‌کند.

متد **fact**: این فاکتوریل عددی را برمی‌گرداند.

متد **pow**: این متد توان  $n$  عددی را برمی‌گرداند.

```
using System;
namespace p6
{
    class Program
    {
        static double
        fact(int n)
        {
            if (n == 1)
                return 1.0;
            else
                return (n * fact(n - 1));
        }
        static double pow(double x, int n)
        {
            if (n == 0)
                return 1.0;
            else
                return (x * pow(x, n - 1));
        }
        static void Main(string[] args)
        {
            int n, i, sign = 1;
            double x, sum = 0;
            Console.Write("Enter x:");
            x = double.Parse(Console.ReadLine());
            Console.Write("Enter n:");
            n = int.Parse(Console.ReadLine());
            for (i = 1; i <= n; i += 2)
```

متد	متغیر	هدف
main	x	ورودی
	n	ورودی
	sum	مجموع
	i	شمارنده
	sign	علامت (یکی در میان مثبت و منفی می‌شود)

## ۶۰۰ برنامه C# با حل آن‌ها ۲۱

```

    {
        sum += sign * pow(x, i) / fact(i);
        sign = -sign;
    }
    Console.WriteLine("Sin({0}) = {1}", x, sum);
    Console.ReadKey();
}
}

```

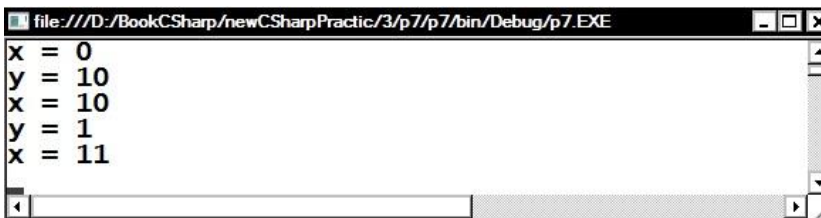


۷. خروجی قطعه برنامه زیر چیست؟ (توضیح برنامه: دو متغیر  $x$  و  $y$  به صورت عمومی تعریف می‌شوند و مقدار آن‌ها صفر است (متغیرهای  $x$  و  $y$  قبل از متد `main` تعریف شده‌اند). با اجرای متد `cout` مقدار  $x = 0$  چاپ می‌گردد. سپس متد `f1` فراخوانی می‌گردد. با فراخوانی متد `f1` متغیر  $y$  به صورت محلی تعریف گردیده، مقدار اولیه ۱۰ به آن تخصیص می‌یابد. سپس با اجرای `cout`، مقدار  $x = 10$  چاپ خواهد شد و مقدار متغیر عمومی  $x$  به ۱۰ تغییر می‌یابد ( $x$   $+= 10$ ). با اجرای `cout`، مقدار  $y = 10$  چاپ خواهد شد و سپس مقدار ۱۱ در  $x$  قرار می‌گیرد ( $x++$ ). پس از برگشت از متد `f1` با اجرای `cout`، ابتدا یک واحد به متغیر عمومی  $y$  اضافه می‌گردد، مقدار ۱ در  $y$  قرار می‌گیرد و مقدار  $y = 1$  چاپ می‌شود. با اجرای آخرین `cout` مقدار  $x = 11$  چاپ می‌شود. سپس، یک واحد به متغیر  $x$  اضافه می‌گردد.

```

using System;
namespace p7
{
    class Program
    {
        static int x, y;
        static void f1()
        {
            int y = 10;
            Console.WriteLine("y = {0}", y);
            x += 10;
            Console.WriteLine("x = {0}", x++);
        }
        static void Main(string[] args)
        {
            Console.WriteLine("x = {0}", x);
            f1();
            Console.WriteLine("y = {0}", ++y);
            Console.WriteLine("x = {0}", x++);
            Console.ReadKey();
        }
    }
}

```



۱. برنامه‌ای که عددی صحیح را از ورودی خوانده، تمام اعداد اول قبل از آن را با استفاده از تعریف زیر تعیین کرده، به خروجی می‌برد. عددی اول است که بر هیچ عدد اول قبل از خودش قابل قسمت نباشد.

وظایف متدها

متد main: متغیر برنامه را تعریف کرده، متد prim را فراخوانی می‌کند.

متد prim: تشخیص می‌دهد عددی اول است یا خیر؟ اگر عدد اول باشد، یک را برمی‌گرداند.

متد	متغیر	هدف
main	n	عدد خوانده شده
	p	آرایه‌ای از اعداد اول
	i	شمارنده
	tedad	تعداد اعداد اول
prim	i	شمارنده

```
using System;
namespace p1
{
    class Program
    {
        static int prim(int[] a, int tedad, int num)
        {
            int i;
            for (i = 0; i < tedad; i++)
                if (num % a[i] == 0)
                    return (0);
            a[tedad] = num;
            return (1);
        }
        static void Main(string[] args)
        {
            int n, i, tedad = 0;
            int[] p = new int[300];
            Console.WriteLine("Enter n:");
            n = Convert.ToInt32(Console.ReadLine());
            for (i = 2; i <= n; i++)
                if (prim(p, tedad, i) == 1) tedad++;
            Console.WriteLine("Primary is ");
            for (i = 0; i < tedad; i++)
                Console.Write(" {0}", p[i]);
            Console.ReadKey();
        }
    }
}
```

۲. برنامه‌ای که شماره دانشجویی تعدادی از دانشجویان را از ورودی خوانده، در آرایه قرار می‌دهد. سپس عناصر آرایه را به روش انتخابی مرتب می‌کند. مرتب‌سازی آرایه به روش انتخابی به این صورت انجام می‌شود: کوچک‌ترین عنصر آرایه پیدا شده، جای آن با عنصر اول آرایه عوض می‌شود. در مرحله بعد بقیه عناصر آرایه برای یافتن کوچک‌ترین عنصر آرایه جستجو می‌شود و جای آن با عنصر دوم آرایه عوض می‌شود. این روند تا مرتب‌سازی کامل آرایه ادامه می‌یابد. پس از مرتب‌سازی نتیجه را در خروجی چاپ می‌کند. این برنامه سه متد دارد که عبارت‌اند از: متدی برای خواندن عناصر آرایه، متدی برای مرتب‌سازی و متدی برای چاپ عناصر آرایه.

وظایف متدها:

## ۶۰۰ برنامه C# با حل آن‌ها ۲۳

متد main: متغیرهای برنامه را تعریف کرده، تعداد دانشجویان را می‌خواند. سپس متدها readarray، select\_sort و writearray را فراخوانی می‌کند.

متد readarray: شماره‌های دانشجویی را می‌خواند و در آرایه قرار می‌دهد.

متد select\_sort: شماره‌های دانشجویی موجود در آرایه را مرتب می‌کند.

متد writearray: شماره‌های دانشجویی موجود در آرایه را چاپ می‌کند.

متد	متغیر	هدف
main	id	آرایه‌ای برای نگهداری شماره دانشجویان
	n	تعداد دانشجویان
	i	شمارنده
readarray	i	شمارنده
select_sort	i, j	شمارنده‌ها
	temp	متغیر کمکی
	max	بزرگ‌ترین عدد
writearray	i	شمارنده

```
using System;
namespace p2
{
    class Program
    {
        static void readarray(int[] id, int n)
        {
            int i;
            for (i = 0; i < n; i++)
            {
                Console.Write("Enter id[{0}];", i + 1);
                id[i] = int.Parse(Console.ReadLine()); ;
            }
        }
        static void select_sort(int[] id, int n)
        {
            int i, j, max, temp;
            for (i = 0; i < n; i++)
            {
                max = i;
                for (j = i + 1; j < n; j++)
                    if (id[max] > id[j]) max = j;
                temp = id[i];
                id[i] = id[max];
                id[max] = temp;
            }
        }
        static void writearray(int[] id, int n)
        {
            int i;
            Console.Write("Sorted output ");
            for (i = 0; i < n; i++)
                Console.Write("\t{0}", id[i]);
        }
        static void Main(string[] args)
        {
            int n;
            int[] id = new int[100];
            Console.Write("Enter n:");
            n = int.Parse(Console.ReadLine()); ;
            readarray(id, n);
            select_sort(id, n);
        }
    }
}
```

```

        writearray(id, n);
        Console.ReadKey();
    }
}

```

۳. برنامه‌ای که تعدادی عدد را از ورودی خوانده، آن‌ها را به‌طور مرتب در آرایه‌ای قرار می‌دهد (دقت داشته باشید که اعداد در موقع گرفتن در آرایه، به‌طور صعودی مرتب شوند). سپس، آرایه مرتب‌شده، را به خروجی می‌برد.

```

using System;
namespace p3
{
    class Program
    {
        static void
args)
    {
        int i,j,l,n,num;
        int[] a = new int[100];
        Console.Write("Enter n:");
        n = int.Parse(Console.ReadLine()); ;
        for (i = 0; i < n; i++)
        {
            Console.Write("Enter a[{0}]:", i+1);
            num = int.Parse(Console.ReadLine()); ;
            for (j = 0; j < i; j++)
                if (a[j] > num)
                    break;
            for (l = i; l > j; l--) a[l] = a[l - 1];
            a[j] = num;
        }
        Console.WriteLine("Result is : ");
        for (i = 0; i < n; i++)
            Console.Write(" {0}", a[i]);
        Console.ReadKey();
    }
}

```

متغیر	هدف
A	آرایه‌ای از اعداد
N	تعداد اعداد
i, j, l	شمارنده‌ها
num	عدد خوانده‌شده

Main(string[]

```
file:///D:/BookCSharp/newCSharpPractic/4/p3/p3/bin/Debug/p3.EXE
Enter n:6
Enter a[1]:12
Enter a[2]:8
Enter a[3]:7
Enter a[4]:4
Enter a[5]:9
Enter a[6]:17
Result is :
      4  7  8  9 12 17
```

۴. برنامه‌ای که دو رشته S۱ و S۲ را از ورودی خوانده، رشته S۱ را در رشته S۲ جستجو می‌کند. در این برنامه، خواندن رشته‌ها توسط متد اصلی و جستجوی رشته توسط متدی انجام می‌شود.

وظایف متدها:

- متد main: دو رشته را خوانده، با فراخوانی متد search\_string، رشته اول را در رشته دوم جستجو می‌کند. سپس، پیغام مناسب چاپ می‌نماید.
- متد search\_string: رشته اول را در رشته دوم جستجو می‌کند. اگر رشته اول در رشته دوم باشد، مکان شروع رشته اول در رشته دوم، وگرنه صفر را برمی‌گرداند.

متد	متغیر	هدف
Main	s۱, s۲	رشته اول
	location	عمل وقوع رشته S۲ در رشته S۱
search_string	i, j	شمارنده‌ها
	k	شمارنده از مکان اولین تساوی

```
using System;
namespace p4
{
    class Program
    {
        static int search_string(char[] s1, char[] s2)
        {
            int i, j, k;
            for (i = 0; i < s2.Length; i++)
            {
                k = i;
                j = 0;
                while(j<s1.Length && k<s2.Length && s1[j]== s2[k])
                {
                    j++;
                    k++;
                }
                if (j == s1.Length)
                    return (i + 1);
                if (j == s1.Length && k == s2.Length)
                    return (i + 1);
            }
            return 0;
        }
        static void Main(string[] args)
        {
            string s1, s2;
            int location;
            Console.Write("Enter string 1:");
            s1 = Console.ReadLine();
```

```

Console.WriteLine("Enter string 2:");
s2 = Console.ReadLine();
char[] ss1 = new char[s1.Length];
s1.CopyTo(0, ss1, 0, ss1.Length);
char[] ss2 = new char[s2.Length];
s2.CopyTo(0, ss2, 0, ss2.Length);
location = search_string(ss1, ss2);
if (location != 0)
    Console.WriteLine("Found in location {0}", location);
else
    Console.WriteLine("Not found ");
Console.ReadKey();
}
}
}

```



۵. برنامه‌ای که با خواندن تعدادی عدد از ورودی، آن‌ها را در آرایه‌ای قرار می‌دهد. سپس، کلیه عناصر آرایه را بر عنصر وسط تقسیم می‌کند. اگر عنصر وسط صفر باشد، بر عنصر بعد از عنصر وسط تقسیم می‌نماید. اگر این عنصر صفر باشد، بر عنصر قبل از عنصر وسط تقسیم می‌کند. اگر این عنصر صفر باشد، برای پیدا کردن عنصری غیر از صفر و انجام تقسیم، به روند قبلی ادامه می‌دهد. اگر همه عناصر آرایه صفر باشند، پیام مناسبی صادر می‌کند. توضیح: پس از قرار دادن عناصر در آرایه، اندیس وسط را محاسبه کرده، عنصر وسط را پیدا می‌نماید. اگر صفر نبود، عمل تقسیم را انجام می‌دهد. این شمارنده به اندیس وسط اضافه می‌شود تا اگر مثبت بود، عنصر بعد از عنصر وسط و اگر منفی بود، عنصر قبل از عنصر وسط مورد بررسی قرار گیرد تا عمل تقسیم انجام شود. اگر عنصر غیر صفری در آرایه پیدا شود، action فراخوانی می‌شود تا تمام عناصر آرایه را بر آن عنصر تقسیم کند. اگر همه عناصر صفر باشند، برنامه پیام مناسبی صادر می‌کند.

متد	متغیر	هدف
main	n	تعداد عناصر آرایه
	a	آرایه‌ای با ۱۰۰ عنصر
	mid	عنصر وسط
	i	شمارنده
	sign	علامت (برای تعیین عنصر قبل و بعد به کار می‌رود)
action	i	شمارنده

```

using System;
namespace p5
{
    class Program
    {
        static void action(int[] a, int loc, int n)
        {
            int i;
            for (i = 0; i < n; i++)
                Console.WriteLine("\n{0} / {1} = {2}", a[i], a[loc],
                    (float)a[i] / a[loc]);
        }
        static void Main(string[] args)
        {
            int i, mid, n, sign = -1;
            int[] a = new int[100];

```

```

Console.Write("Enter n:");
n = int.Parse(Console.ReadLine());
mid = n / 2;
for (i = 0; i < n; i++)
{
    Console.Write("Enter a[{0}]:", i + 1);
    a[i] = int.Parse(Console.ReadLine());
}
for (i = 0; i <= mid; i++)
{
    if (a[mid + (sign * i)] != 0)
    {
        action(a, mid + (sign * i), n);
        break;
    }
    if (sign == 1) i--;
    sign = -sign;
}
if (i > mid)
    Console.Write("All element are zero");
Console.ReadKey();
}
}

```

```

file:///D:/BookCSharp/newCSharpPractic/4/p5/p5/bin/Debug/p5.EXE
Enter n:8
Enter a[1]:9
Enter a[2]:8
Enter a[3]:9
Enter a[4]:0
Enter a[5]:0
Enter a[6]:0
Enter a[7]:0
Enter a[8]:6

9 / 9 = 1
8 / 9 = 0.8888889
9 / 9 = 1
0 / 9 = 0
0 / 9 = 0
0 / 9 = 0
0 / 9 = 0
6 / 9 = 0.6666667_

```

۶. کارخانه‌ای دارای ۵ ردیف شغلی است که به کارمندان در مقابل ۴۰ ساعت کار در هفته، ماهانه مطابق جدول زیر حقوق ثابتی پرداخت می‌شود. در صورتی که کارمندی بیش از ۱۶۰ ساعت در ماه کار کند اضافه‌کاری به ازای هر ساعت مطابق به آنچه که در جدول مشخص شد، پرداخت می‌گردد. اگر کارمندی کم‌تر از ۱۶۰ ساعت در ماه کار کند، به ازای هر ساعت مبلغی مطابق جدول از حقوق وی کسر می‌شود.

برنامه‌ای که شماره کارمندی، ردیف شغلی و ساعت کارکرد در ماه برای هر کارمند را خوانده، دریافتی آن‌ها را چاپ می‌کند. برای خاتمه برنامه به جای شماره کارمندی ۹۹۹- وارد می‌شود.

ردیف شغلی	نوع تخصص	حقوق ثابت	هر ساعت اضافه‌کاری	هر ساعت کم‌کاری
۰	مهندس ارشد	۱۰۰۰ و ۱۰۰۰	۵۰۰۰	۶۰۰۰
۱	مهندس ساده	۸۰۰ و ۱۰۰۰	۴۰۰۰	۵۰۰۰
۲	تکنسین	۶۰۰ و ۱۰۰۰	۳۰۰۰	۴۰۰۰
۳	کارگر ماهر	۵۰۰ و ۱۰۰۰	۲۵۰۰	۳۰۰۰
۴	کارگر ساده	۴۶۰ و ۱۰۰۰	۲۰۰۰	۲۵۰۰

متغیر	هدف
table	جدولی با ۵ سطر و ۳ ستون
code	شماره کارمندی
radif	ردیف شغلی
hh	ساعات کارکرد
t	ساعات تأخیر
e	ساعات اضافه کار
t_mablag	مبلغ تأخیر
base	پایه
e_mablag	مبلغ اضافه کار
pay	قابل پرداخت

```
using System;
namespace p6
{
    class Program
    {
        static void Main(string[] args)
        {
            long[, ] table = new long[5, 3]
                {{1000000, 5000, 6000},
                {800000, 4000, 5000},
                {600000, 3000, 4000},
                {500000, 2500, 3000},
                {460000, 2000, 2500}};
            int code, radif, hh, t, e;
            long t_mablag, e_mablag, pay, base1;
            for ( ; ; )
            {
                t = 0;
                e = 0;
                Console.Write("Enter code:");
                code = int.Parse(Console.ReadLine());
                if (code == -999) break;
                Console.Write("Enter radif:");
                radif = int.Parse(Console.ReadLine());
                Console.Write("Enter hour:");
                hh = int.Parse(Console.ReadLine());
                if (radif > 4 || radif < 0)
                {
                    Console.Write("Enter radif between 0 to 4");
                    continue;
                }
                if (hh > 160)
                    e = hh - 160;
                else
                    t = 160 - hh;
                base1 = table[radif, 0];
                e_mablag = table[radif, 1] * e;
                t_mablag = table[radif, 2] * t;
                pay = base1 + e_mablag - t_mablag;
                Console.WriteLine("Code is {0}\thour is {1}", code, hh);
                Console.WriteLine("\nbase is {0}\te_mablag is {1}\tt_mablag is {2}\tpay{3}", base1,
                    e_mablag, t_mablag, pay);
            }
        }
    }
}
```

## ۶۰۰ برنامه C# با حل آن‌ها ۲۹

```

    }
}
}

```

```

file:///D:/BookCSharp/newCSharpPractic/4/p6/p6/bin/Debug/p6.EXE
Enter code:1
Enter radif:1
Enter hour:12
Code is 1      hour is 12      e_mablag is 0      t_mablag is 740000      Pay60000
base is 800000
Enter code:2
Enter radif:2
Enter hour:50
Code is 2      hour is 50      e_mablag is 0      t_mablag is 440000      Pay160000
base is 600000
Enter code:3
Enter radif:3
Enter hour:20
Code is 3      hour is 20      e_mablag is 0      t_mablag is 420000      Pay80000
base is 500000
Enter code:4
Enter radif:4
Enter hour:55
Code is 4      hour is 55      e_mablag is 0      t_mablag is 262500      Pay197500
base is 460000
Enter code:-999_

```

۱. برنامه‌ای که کلاس Circle را طوری تغییر می‌دهد تا دو سازنده داشته باشد. سازنده اول، سازنده پیش فرض را پیاده‌سازی می‌کند و سازنده دوم شعاع دایره را به عنوان پارامتر دریافت می‌کند و در فیلد شعاع (r) قرار می‌دهد.

مراحل طراحی و اجرا

۱. پروژه جدیدی ایجاد کنید.

۲. کلاس Circle را به پروژه اضافه کرده، دستورات آن را به صورت زیر تغییر دهید:

```

using System;
namespace p1
{
    class Circle
    {
        public const double PI = Math.PI;
        private int r;
        public int R
        {
            get
            {
                return r;
            }
            set
            {
                if (value > 0) r = value;
                else r = 0;
            }
        }
        public Circle()
        {
            R = 0;
        }
        public Circle(int r1)
        {
            R = r1;
        }
    }
}

```

```

        public double Area()
        {
            return PI * r * r;
        }
        public double Perime()
        {
            return 2 * PI * r;
        }
    }
}

```

۳. به کلاس Program برگردید و دستورات آن را به صورت زیر تغییر دهید:

```

using System;
namespace p1
{
    class Program
    {
        static void Main(string[] args)
        {
            Circle c1 = new Circle();
            Console.WriteLine("Radis is {0}\tArea is {1}\tPerime
                is {2}", c1.R, c1.Area(), c1.Perime());
            Console.Write("Enter Radius:");
            Circle c2=new Circle(Convert.ToInt32(Console.ReadLine()));
            Console.WriteLine("Radis is {0}\tArea is {1}\tPerime
                is {2}", c2.R, c2.Area(), c2.Perime());
            Console.ReadKey();
        }
    }
}

```

در این دستورات ابتدا نمونه C۱ ایجاد می شود. برای ایجاد نمونه C۱ سازنده بدون پارامتر فراخوانی می شود تا شعاع برابر صفر خواهد شد. در ادامه شعاع، محیط و مساحت را برای نمونه C۱ نمایش می دهد. در ادامه شعاع را از کاربر درخواست می کند. با شعاع وارد شده توسط کاربر، نمونه C۲ را ایجاد می کند که برای ایجاد نمونه C۲ سازنده با یک پارامتر فراخوانی می کند و در پایان شعاع، محیط و مساحت نمونه C۲ را نمایش می دهد.

۴. پروژه را ذخیره و اجرا کنید تا خروجی زیر را مشاهده نمایید:

```

file:///D:/BookCSharp/newCSharpPractic/5/p1/p1/bin/Debug/p1.EXE
Radis is 0      Area is 0      Perime is 0
Enter Radius:5
Radis is 5      Area is 78.5398163397448      Perime is 31.4159265358979

```

۲. برنامه ای که تعداد اشیاء ایجاد شده یک کلاس را شمارش می کند و نمایش می دهد.

مراحل طراحی و اجرا

۱. پروژه جدیدی ایجاد کنید.

۲. کلاسی به نام Counter ایجاد کرده، دستورات آن را به صورت زیر تغییر دهید:

```

using System;
namespace p2
{
    class Counter
    {
        public static int countObject = 0;
    }
}

```

### ۶۰۰ برنامه C# با حل آن‌ها ۳۱

```
public Counter()
{
    countObject++;
}
~Counter()
{
    --countObject;
}
}
```

دارای اعضای زیر است: Counter کلاس

✚ **عضو استاتیک** countObject، تعداد نمونه‌های (اشیا) ایجادشده کلاس را نگهداری می‌کند.

✚ **سازنده** Counter()، به countObject یک واحد اضافه می‌کند تا تعداد نمونه‌های ایجادشده از کلاس در این فیلد قرار گیرد.

✚ **مخرب** Counter()، از فیلد countObject یک واحد کم می‌کند تا با تخریب (حذف) شی‌ایی از کلاس از فیلد countObject یک واحد کم شود. همان‌طور که در تعریف مخرب می‌بینید، علامت (~) قبل از نام کلاس آمده است. البته در مخرب نباید سطح دسترسی public یا private را تعیین کنید.

۳. به کلاس Program بروید و دستورات آن را به‌صورت زیر تغییر دهید:

```
using System;
namespace p2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Count Object is {0}", Counter.countObject);
            Counter c1 = new Counter();
            Counter c2 = new Counter();
            Console.WriteLine("Count Object is {0}", Counter.countObject);
            Console.ReadKey();
        }
    }
}
```

دستور اول داخل متد Main، تعداد نمونه‌های ایجادشده کلاس Counter را نمایش می‌دهد که صفر است.

دستورات دوم و سوم، دو نمونه به نام‌های c1 و c2 از نوع کلاس Counter ایجاد می‌کنند و دستور چهارم، تعداد نمونه‌های ایجادشده کلاس Counter (یعنی، ۲) را نمایش خواهد داد.

۴. پروژه را ذخیره و اجرا کنید تا خروجی زیر را ببینید:



۳. برنامه‌ای که نام کاربر و کلمه عبور را دریافت می‌کند. اگر نام کاربر fanavarienovin و کلمه عبور ۱۲۳۴۵۶ باشد، پیغام Correct، وگرنه پیغام Not Correct را نمایش می‌دهد.

توضیح: این برنامه از کلاسی به نام User استفاده می کند که دو فیلد static به نام های userName و Password دارد و متدی به نام isCorrect() دارد تا تعیین کند، نام کاربر و کلمه عبور صحیح وارد شده است یا خیر.

مراحل طراحی و اجرا

۱. پروژه جدیدی ایجاد کنید.

۲. کلاسی User را به برنامه اضافه کرده، دستورات آن را به صورت زیر تغییر دهید:

```
using System;
namespace p3
{
    class User
    {
        private static string UserName = "fanavarienovin";
        private static string Password = "1234556";
        public static bool isCorrect(string userName,string password)
        {
            return (userName == User.UserName && password == User.Password);
        }
    }
}
```

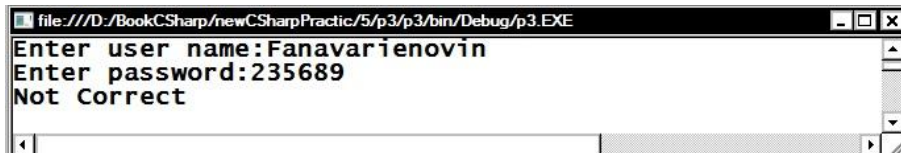
۳. به کلاسی Program بروید و دستورات آن را به صورت زیر تغییر دهید:

```
using System;
namespace p3
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Enter user name:");
            string userName = Console.ReadLine();
            Console.WriteLine("Enter password:");
            string password = Console.ReadLine();
            if (User.isCorrect(userName, password) == true)
                Console.WriteLine("Correct");
            else
                Console.WriteLine("Not Correct");
            Console.ReadKey();
        }
    }
}
```

دستورات اول تا چهارم درون متد Main، با پیغام هایی، نام کاربر و کلمه عبور را خوانده، در متغیرهای userName و Password قرار می دهند و دستور پنجم (if)، متد isCorrect() را با دو پارامتر userName و Password فراخوانی می کند. دقت کنید در هنگام فراخوانی از نام کلاس (یعنی، User) استفاده نموده است (نمونه ای از آن ایجاد نکرد).

۴. پروژه را ذخیره و اجرا کنید تا نمونه خروجی زیر را مشاهده نمایید:

## ۶۰۰ برنامه C# با حل آن‌ها ۳۳



۴. برنامه‌ای که شعاع کره‌ای را دریافت کرده مساحت و حجم کره را محاسبه می‌نماید و نمایش می‌دهد (در این برنامه از کلاسی به نام Sphere استفاده می‌کند که فیلد شعاع (r) آن فقط خواندنی می‌باشد). حجم و مساحت کره با فرمول‌های زیر محاسبه می‌شود:

$$\text{حجم کره} = \frac{4}{3} \pi r^3 = 3,1415 \times r^3 \quad \text{مساحت کره} = 4 \pi r^2 = 3,1415 \times r^2$$

مراحل طراحی و اجرا

۱. پروژه جدیدی ایجاد کنید.

۲. کلاس Sphere را به پروژه اضافه کرده، دستورات آن را به صورت زیر تغییر دهید:

```
using System;
namespace Cho_۷ {
    class Sphere {
    using System;
    namespace p4
    {
        class Sphere
        {
            public const double PI = 3.1415;
            public readonly int r;
            public Sphere(int r)
            {
                if (r > 0) this.r = r;
                else this.r = 0;
            }
            private long Pow(int r, int n)
            {
                long p = 1;
                for (int i = 1; i <= n; i++) p *= r;
                return p;
            }
            public double Volume()
            {
                return (4.0 / 3 * Pow(r, 3) * PI);
            }
            public double Area()
            {
                return (4.0 * Pow(r, 2) * PI);
            }
        }
    }
}
```

این کلاس دارای اعضای زیر می‌باشد:

🚩 ثابت PI، مقدار ۳,۱۴۱۵ را نگهداری می‌کند.

🚩 فیلد فقط خواندنی r، شعاع کره را نگهداری می‌کند.

🚩 سازنده Sphere، مقدار r را به عنوان پارامتر دریافت کرده در فیلد فقط خواندنی شعاع (r) قرار می‌دهد (چون نام متغیر با نام فیلد یکی می‌باشد از اشاره گر this استفاده شده است).

🚩 متد Pow()، پارامتر r و n را گرفته،  $r^n$  را برمی‌گرداند.

متد **Volume()** حجم کره را حساب کرده، برمی گرداند.

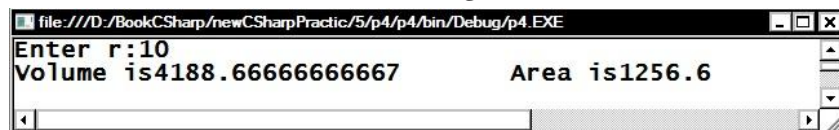
متد **Area()** مساحت کره را محاسبه کرده، برمی گرداند.

۳. به کلاس Program بروید و دستورات آن را به صورت زیر تغییر دهید:

```
using System;
namespace p4
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Enter r:");
            Sphere s = new Sphere(Convert.ToInt32(Console.ReadLine()));
            Console.WriteLine("Volume is{0}\tArea is{1}",s.Volume(),s.Area());
            Console.ReadKey();
        }
    }
}
```

دستور اول داخل **Main()**، یک پیغام نمایش می دهد تا کاربر شعاع کره را وارد کند. دستور دوم با شعاع دریافتی یک نمونه به نام **s** از کلاس **Sphere** ایجاد می کند و دستور سوم، مساحت و حجم کره را محاسبه کرده، نمایش می دهد.

۴. پروژه را ذخیره و اجرا کنید تا نمونه خروجی زیر را مشاهده نمایید:



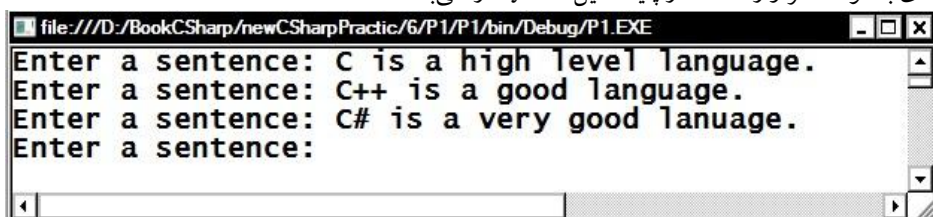
۱. برنامه ای که چند جمله را دریافت کرده، در فایل **۱.XT** در درایو **D** پوشه **Data** می نویسد.

```
using System;
using System.IO;
namespace P1
{
    class Program
    {
        static void Main(string[] args)
        {
            string sentence = " ";
            StreamWriter myfile;
            myfile = File.CreateText(@"D:\data\1.txt");
            do
            {
                Console.Write("Enter a sentence: ");
                sentence = Console.ReadLine();
                if (sentence.Length != 0)
                {
                    myfile.WriteLine(sentence);
                }
            } while (sentence.Length != 0);
            myfile.Close();
        }
    }
}
```

## ۶۰۰ برنامه C# با حل آن‌ها ۳۵

```
}  
}  
}
```

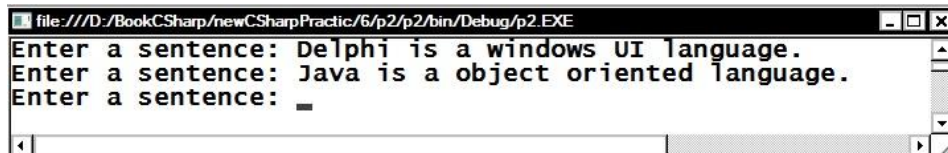
این برنامه ابتدا متغیر رشته‌ای sentence را برای دریافت جملات تعریف می‌کند، سپس متغیر myfile را با نوع StreamWriter تعریف می‌نماید و فایل متنی ۱.txt را در پوشه Data درایو D ایجاد می‌کند با متد (createText). در ادامه در یک حلقه با پیغام مناسب جمله‌ای را خوانده، در متغیر sentence قرار می‌دهد و چنانچه طول رشته sentence صفر نباشد، آن را در myfile می‌نویسد و این روند را ادامه می‌دهد تا کاربر رشته‌ای با طول صفر وارد کند. در پایان فایل myfile را می‌بندد.



۲. برنامه‌ای که چند جمله را می‌خواند به انتهای فایل متنی ۱.txt در پوشه Data درایو D اضافه می‌کند.

```
using System;  
using System.IO;  
namespace p2  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            StreamWriter myFile = File.AppendText(@"d:\data\1.txt");  
            string line="";  
            do  
            {  
                Console.Write("Enter a sentence: ");  
                line = Console.ReadLine();  
                if (line != "")  
                    myFile.WriteLine(line);  
            }  
            while (line != "");  
            myFile.Close();  
        }  
    }  
}
```

در این برنامه ابتدا متغیر myfile را با نوع StreamWriter تعریف کرد و فایل متنی ۱.txt موجود در پوشه درایو D را طوری باز می‌کند تا بتوان اطلاعاتی را به انتهای آن اضافه نمود. سپس متغیر line را به صورت رشته‌ای تعریف کرده و در حلقه تکرار با پیغام مناسب جمله‌ای را خوانده و در line قرار می‌دهد و اگر line خالی نباشد، آن را به انتهای فایل myfile اضافه می‌کند و اگر line خالی باشد، حلقه خاتمه می‌یابد و myfile را می‌بندد.

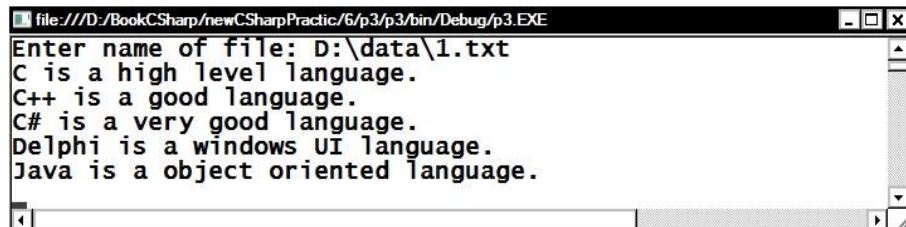


۳. برنامه‌ای که نام یک فایل متنی را از کاربر دریافت کرده، اطلاعات آن را بر روی صفحه نمایش نشان می‌دهد.

```
using System;
using System.IO;
namespace p3
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Enter name of file: ");
            string fileName = Console.ReadLine();
            StreamReader myfile;
            try
            {
                myfile = File.OpenText(fileName);
                string line = " ";
                do
                {
                    line = myfile.ReadLine();
                    if (line != null)
                    {
                        Console.WriteLine(line);
                    }
                } while (line != null);
            }
            catch (Exception e)
            {
                Console.WriteLine("Error in file.");
            }
            Console.ReadKey();
        }
    }
}
```

در این برنامه ابتدا با یک پیغام مناسب نام فایل را خوانده، در متغیر رشته‌ای `fileName` قرار می‌دهد. سپس متغیر `myfile` را با نوع `FileReader` تعریف می‌کند تا بتوان اطلاعات آن فایل را خواند. در ادامه فایل را به صورت خواندنی باز کرده و دستگیره آن را در متغیر `myfile` قرار می‌دهد. در پایان، رشته `line` را تعریف کرده و خط به خط از `myfile` می‌خواند و بر روی صفحه نمایش نشان می‌دهد. این کار را تا زمانی ادامه می‌دهد که `line` برابر `null` شود. یعنی خطی در فایل موجود نباشد. این دستورات را بین `try` و `catch` قرار می‌دهد تا چنانچه خطای رخ دهد، در بخش `catch` عبارت `Error in file` را نمایش دهد. در خاتمه فایل را می‌بندد.

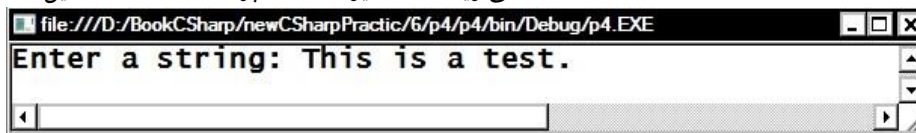
## ۶۰۰ برنامه C# با حل آن‌ها ۳۷




۴. برنامه‌ای که یک رشته را خوانده با استفاده از کلاس Logger در فایل log.txt در پوشه Data درایو D می‌نویسد.


```
using System;
namespace p4
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Enter a string: ");
            string s = Console.ReadLine();
            Logger.Write(@"D:\data\log.txt", s);
            Console.ReadKey();
        }
    }
}
```

Logger قرار می‌دهد. سپس با استفاده از کلاس S این برنامه با پیغام مناسب رشته‌ای را خوانده و در متغیر می‌نویسد. D درایو Data در پوشه log.txt را در فایل S رشته



۵. برنامه‌ای که نام یک فایل متنی را خوانده و اطلاعات فایل را بر روی صفحه‌نمایش نشان می‌دهد. چنانچه تعداد خطوط فایل بیش از ۲۲ سطر باشد، پس از نمایش هر ۲۲ سطر یک مکث موقت می‌کند تا کاربر کلیدی را فشار دهد.

**متد**  `showData()` نام فایلی که می‌خواهد نمایش دهد را با پارامتر `urlFile` دریافت می‌کند و اطلاعات آن فایل را ۲۲ خط ۲۲ خط نمایش می‌دهد. برای این منظور، ابتدا فایل متنی که نام آن در `urlFile` قرار دارد را به صورت خواندنی باز کرده و اشاره‌گر آن در متغیر `myFile` با نوع `FileReader` قرار می‌دهد و متغیر `count` (تعداد خطوط نمایش داده شده) را تعریف می‌کند و مقدار صفر را در آن قرار می‌دهد. در ادامه با استفاده از یک حلقه `while` ..... `do` خط به خط از فایل `myFile` خوانده و بر روی صفحه‌نمایش نشان می‌دهد. اگر تعداد خطوط نمایش داده شده ۲۲ شود (یعنی ۲۲٪ `count` برابر صفر گردد، یک پیغام `Press any key to continue....` را نمایش می‌دهد و این روند را تا زمانی ادامه می‌دهد که تمام خطوط فایل خوانده شود. در پایان، فایل `myFile` را می‌بندد.

**متد**  `main()` با پیغام مناسبی نام فایل را خوانده در متغیر `fileName` قرار می‌دهد و با فراخوانی `ShowData()` اطلاعات فایلی که نام آن در `fileName` است را به صورت صفحه به صفحه نمایش می‌دهد.

```
using System;
```

```
using System.IO;
namespace p5
{
    class Program
    {
        static public void ShowData(string urlFile)
        {
            StreamReader myFile = new StreamReader(urlFile);
            string line;
            int count = 0;
            do
            {
                line = myFile.ReadLine();
                if (line != null)
                {
                    if (count != 0 && count % 22 == 0)
                    {
                        Console.WriteLine("Perss any key to continue ... ");
                        Console.ReadKey();
                    }
                    if (line.Length > 79)
                        line = line.Substring(0, 79);
                    Console.WriteLine(line);
                }
                count++;
            } while (line != null);
            myFile.Close();
        }
        static void Main(string[] args)
        {
            Console.Write("Enter name of file: ");
            string fileName = Console.ReadLine();
            ShowData(fileName);
        }
    }
}
```

➤ برنامه‌ای که مقدار  $x$  را از ورودی خوانده، حاصل عبارت مقابل را محاسبه می‌کند:

$$y = \frac{1}{x^2 + x + 1}$$

➤ برنامه‌ای که دو عدد را از ورودی می‌خواند (عدد اول مقاومت یک مدار الکتریکی و عدد دوم جریان آن را نمایش می‌دهد). با فرمول مقابل ولتاژ را محاسبه کرده، نمایش می‌دهد:

$$\text{جریان} * \text{مقاومت} = \text{ولتاژ}$$

➤ مجموع مقاومت‌های  $R_1$ ،  $R_2$  و  $R_3$  که به‌طور موازی به هم متصل‌اند، با فرمول زیر محاسبه می‌شود:

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} = \frac{1}{R} = \frac{R_2 * R_3 + R_1 * R_3 + R_1 * R_2}{R_1 * R_2 * R_3}$$

برنامه‌ای که سه مقاومت را از ورودی خوانده، مقاومت کل را محاسبه می‌کند.

➤ برنامه‌ای که وزن کالایی را برحسب کیلوگرم دریافت می‌نماید و وزن آن را برحسب گرم نمایش می‌دهد. هر کیلوگرم برابر ۱۰۰۰ گرم است.

➤ اگر کارمندی به اندازه ۱۵ درصد حقوق ماه گذشته خود پاداش بگیرد، برنامه‌ای که حقوق کارمند را خوانده، پاداش او را محاسبه نموده، نمایش می‌دهد.

➤ برنامه‌ای که ارتفاع و قاعده مثلث را از ورودی خوانده، مساحت آن را محاسبه کرده، به خروجی می‌برد.

$$\text{قاعده} * \text{ارتفاع} \times 0,5 = \text{مساحت مثلث}$$

➤ در شرکتی، سالانه ۱۵۰ خودکار، ۵۰ بسته کاغذ A4 مصرف می‌شود. در پایان سال این شرکت می‌خواهد بداند در سال آینده چقدر برای این بخش از تجهیزات اداری، افزایش هزینه دارد. برنامه‌ای که قیمت این اقلام را در امسال از ورودی دریافت می‌نماید، نرخ تورم در سال آینده را نیز می‌گیرد، هزینه اضافه شده شرکت را در این بخش محاسبه می‌کند و به خروجی می‌برد. تورم به صورت درصد وارد می‌گردد که برنامه باید آن را به مقدار اعشاری تبدیل کند. به عنوان مثال، اگر تورم را ۵,۶ وارد کنیم، برنامه باید از ۰/۰۵۶ استفاده کند.

➤ یک دوچرخه‌سوار با سرعت  $x$  کیلومتر بر ساعت شروع به حرکت می‌کند و پس از  $n$  دقیقه سرعت آن به  $k$  کیلومتر در ساعت می‌رسد. برنامه‌ای که با استفاده از فرمول زیر شتاب او را محاسبه کرده، نمایش می‌دهد:

$$\text{شتاب} = \frac{60 * (\text{سرعت اولیه } x) + (\text{سرعت نهایی } k)}{n(\text{زمان})}$$

➤ برنامه‌ای که مصرف (کیلومتر به لیتر) و گنجایش باک بنزین (لیتر) اتومبیلی را از ورودی می‌خواند و تعیین می‌کند که با یک باک بنزین اتومبیل چه مسافتی را می‌تواند بپیماید.

➤ برنامه‌ای که دو عدد را از کاربر دریافت می‌کند و حاصل جمع، ضرب، تقسیم، تفریق و باقی‌مانده تقسیم صحیح آن‌ها را چاپ می‌کند.

➤ برنامه‌ای که دو عدد صحیح را از کاربر دریافت کرده، عدد بزرگ‌تر و سپس عبارت `is larger than` و عدد کوچک‌تر را چاپ می‌کند. اگر این دو عدد برابر باشند، عبارت `"These number are equal"` را چاپ می‌کند.

- برنامه‌ای که سه عدد را از صفحه کلید می‌گیرد و حاصل جمع، میانگین، حاصل ضرب، بزرگ‌ترین و کوچک‌ترین عدد را چاپ می‌کند.
- برنامه‌ای که دو عدد را از صفحه کلید می‌گیرد و تعیین می‌کند که آیا عدد اول بر عدد دوم بخش پذیر است یا نه؟
- برنامه‌ای که دو عدد را خوانده، تعیین می‌کند آیا عدد اول مضربی از عدد دوم است یا خیر.
- برنامه‌ای که پنج عدد صحیح را خوانده، بزرگ‌ترین و کوچک‌ترین عدد را نمایش می‌دهد.
- با استفاده از امکاناتی که در این فصل دیدید برنامه‌ای که توان ۲، توان ۳، و عدد ۰ تا ۱۰ را به صورت زیر جدول بندی می‌نماید؟ (توضیح: همان طور که مشاهده نمودید، تعداد خطوط این برنامه زیاد شده است که با بیان حلقه‌های تکرار در فصل بعد، تعداد دستورات این قبیل برنامه‌ها کاهش خواهد یافت).
- برنامه‌ای که یک عدد صحیح پنج رقمی را خوانده، ارقام آن را جدا می‌کند و هر یک از ارقام را با سه فاصله بین آن‌ها چاپ می‌کند. به عنوان مثال، اگر کاربر عدد ۴۲۳۳۹ را وارد کند، خروجی به صورت زیر باشد:  

$$\text{Result is } \quad 4 \quad 2 \quad 3 \quad 3 \quad 9$$
- هر تن ۳۵۲۷۳٫۹۲ اونس است. برنامه‌ای که وزن یک محموله را به اونس گرفته، به تن تبدیل می‌کند. برنامه به کاربر اجازه می‌دهد تا هر زمان می‌خواهد این محاسبه را تکرار کند.
- هر لیتر معادل ۰٫۲۶۴۱۷۹ گالن است. برنامه‌ای که میزان بنزین مصرفی اتومبیل کاربر را برحسب لیتر و مسافت طی شده آن را به مایل گرفته، سپس مصرف به ازای هر مایل به گالن را نمایش می‌دهد.
- برنامه‌ای که عددی را خوانده، مربع و مکعب آن را نمایش می‌دهد.
- برنامه‌ای که دو ضلع موازی و ارتفاع (a, b و h) ذوزنقه را دریافت کرده، مساحت آن را با فرمول زیر محاسبه می‌کند:  

$$s = h * (a + b) / 2$$
- برنامه‌ای که شعاع دایره را خوانده، محیط، مساحت و قطر آن را نمایش می‌دهد (محیط، مساحت و قطر دایره به صورت زیر حساب می‌شوند):  

$$\pi * \text{شعاع} * 2 = \text{محیط دایره}$$

$$\begin{aligned} \text{مساحت دایره} &= \pi * (\text{شعاع})^2 \\ \text{شعاع} * 2 &= \text{قطر دایره} \end{aligned}$$

➤ برنامه‌ای که عددی را به سال خوانده، تعیین می‌کند چند ماه، چند روز و چند ثانیه است.

$$\begin{aligned} \text{سال} &= 365,25 * \text{روز} \\ 60 * 24 * \text{روز} &= \text{ثانیه} \quad 30 / \text{روز} = \text{ماه} \end{aligned}$$

➤ برنامه‌ای که m و n را خوانده، حاصل عبارات زیر را محاسبه کند:

$$a = m^2 - n^2 \quad b = 2 * m * n \quad c = m^2 + n^2$$

➤ برنامه‌ای که دو عدد را خوانده، در متغیرهای a و b قرار می‌دهد، سپس حاصل جمع، تفریق، ضرب، تقسیم،  $a^2 + b^2$  و  $a^3 + b^3$  را نمایش می‌دهد.

➤ برنامه‌ای که یک عدد سه‌رقمی را خوانده، اگر مجموع رقم‌های اول و سوم برابر رقم دوم باشد، "Yes"، وگرنه "No" را چاپ می‌کند.

➤ فرض کنید هر صفحه کتاب ۳۰ سطر (خط) است و هر خط ۸۰ بایت حافظه اشغال می‌کند. برنامه‌ای که تعداد صفحات یک کتاب و ظرفیت یک فلش را به گیگابایت خوانده، تعیین می‌کند که این فلش چند تا از این کتاب‌ها را می‌تواند در خودش ذخیره کند (گیگابایت برابر با  $1024^3$  بایت است).

➤ برنامه‌ای که یک عدد چهاررقمی را خوانده، اگر حاصل ضرب رقم‌های اول و چهارم، برابر حاصل جمع ارقام دوم و سوم باشد، "Yes"، وگرنه "No" را نمایش می‌دهد.

➤ برنامه‌ای که تعداد کالا و قیمت هر کالا را خوانده، مبلغ فروش را نمایش می‌دهد (مبلغ فروش برابر با تعداد کالا \* قیمت کالا است).

➤ برنامه‌ای که سن‌تان را به سال خوانده، مشخص کند چند دقیقه زندگی کرده‌اید (هر سال ۳۶۵,۲۵ روز است و هر روز  $24 \times (60)^2$  دقیقه می‌باشد).

➤ برنامه‌ای که مقداری را به مایل و فوت دریافت کرده، به متر و کیلومتر تبدیل می‌کند و نمایش می‌دهد (هر فوت ۰,۳۰۴۸ متر و هر مایل ۱۶۰۹,۳۵ متر است).

➤ برنامه‌ای که x و y را از ورودی خوانده، حاصل عبارت مقابل را نمایش می‌دهد:

$$Z = x++ * y / x++;$$

➤ برنامه‌ای که یک حرف انگلیسی کوچک را خوانده، به حرف بزرگ تبدیل کرده و نمایش می‌دهد (فاصله بین حرف بزرگ و کوچک ۳۲ کاراکتر است. یعنی، چنانچه از یک حرف کوچک ۳۲ تا کم کنیم به حرف بزرگ تبدیل می‌شود).

➤ برنامه‌ای که یک کاراکتر را خوانده و تعیین می‌کند که بین ۰ تا ۹ است یا خیر؟

➤ برنامه‌ای که  $x$  و  $y$  را خوانده، حاصل عبارت مقابل را نمایش می‌دهد:

$$z = x^3 + 2x^2 + 3y - 5$$

➤ برنامه‌ای که نمره‌ای دانشجویی را دریافت کرده، اگر نمره کم‌تر از ۱۰ بود، کلمه "Failed"، وگرنه کلمه "Passed" را نمایش می‌دهد (در این برنامه متغیر grade، نمره دانشجو است).

➤ برنامه‌ای که دو عدد را خوانده، بدون استفاده از متغیر کمکی محتوی آن‌ها را تعویض می‌کند.

➤ برنامه‌ای که یک حرف کوچک را خوانده، با استفاده از عملگر بیتی آن را به حرف بزرگ تبدیل می‌کند (کد اسکی حروف کوچک ۳۲ واحد از حروف بزرگ بیش‌تر است). برای این منظور، اگر کاراکتر حرف کوچک باشد، آن را با مقدار (۳۲ - ۲۵۵) بیت به بیت و (&) بیتی می‌نماییم تا تبدیل به حرف بزرگ شود.

➤ برنامه‌ای که قاعده و ارتفاع مثلث را خوانده، مساحت مثلث را محاسبه می‌کند (مساحت مثلث برابر با ارتفاع ضرب‌در نصف قاعده است).

➤ پروژه برنامه‌نویسی ۱: برنامه‌ای که ۵ عدد را خوانده، بزرگ‌ترین و کوچک‌ترین عدد را نمایش می‌دهد.

➤ پروژه برنامه‌نویسی ۲: برنامه‌ای که یک عدد ۵ رقمی را خوانده، ارقام عدد را بافاصله نمایش می‌دهد.

## ➤ ساختار تصمیم و حلقه تکرار

- برنامه‌ای که سه مقدار را از ورودی خوانده، آن‌ها را به ترتیب صعودی در خروجی چاپ می‌کند.
- برنامه‌ای که شماره کارمندی، ساعت کارکرد و دستمزد ساعتی کارکنان موسسه‌ای را خوانده، حقوق آن‌ها را محاسبه می‌کند. اگر کارمندی بیش از ۴۰ ساعت کارکرده باشد، اضافه‌کار به او تعلق می‌گیرد. به ازای هر ساعت کاری،  $\frac{3}{2}$  دستمزد ساعتی به‌عنوان اضافه‌کاری پرداخت می‌شود.
- یکی از مسائل مهم شرکت‌ها و ادارات پیش‌بینی بودجه چند سال بعد است. برنامه‌ای که قیمت کالا (ارزش پروژه) و نرخ تورم کالا را گرفته، قیمت کالا را برای چند سال بعد پیش‌بینی می‌نماید (برنامه تعداد سال‌هایی که قیمت کالا باید پیش‌بینی شود، از کاربر دریافت می‌نماید).
- برنامه‌ای که مجموع ده جمله اول سری زیر را محاسبه و چاپ می‌کند.
 
$$s = \frac{1}{x} - \frac{1}{x+2x^2} + \frac{1}{x+2x^2+3x^3} - \frac{1}{x+2x^2+3x^3+4x^4} + \dots$$
- برنامه‌ای که مجموعه‌ای از اعداد دوتایی صحیح را خوانده، حاصل ضرب آن‌ها را بدون استفاده از عملگر ضرب نمایش می‌دهد. چنانچه کاربر به‌جای دو عدد، صفر وارد کند، برنامه خاتمه یابد.
- برنامه‌ای که عددی را خوانده، مشخص می‌کند آیا عدد متقارن است یا خیر. نمونه‌های از اعداد متقارن عبارت‌اند از: ۱۲۴۲۱ و ۳۲۴۲۳. (توضیح: چنانچه عددی برابر با مغلوبش باشد، متقارن است. پس، ابتدا مغلوب عدد را محاسبه کرده، با خودش مقایسه می‌کنیم).
- برنامه‌ای که  $m$  و  $n$  دو عدد صحیح و مثبت را خوانده، با استفاده از عملگر  $+$ ،  $m^n$  را محاسبه و چاپ می‌نماید.
- برنامه‌ای که تعدادی عدد را خوانده، هر یک از اعداد را که بر ۹ بخش پذیر باشد، در خروجی چاپ می‌کند (از روش مجموع ارقام استفاده کنید).
- برنامه‌ای که اطلاعات تعدادی کارمند از قبیل شماره کارمندی، سال استخدام، کد جنسیت، (۰: زن، ۱: مرد) پایه استخدام (۹-۱) و حقوق را خوانده، موارد زیر را در خروجی چاپ می‌کند.

الف. تعداد کارمندان مرد      ب. تعداد کارکنان زن      ج. میانگین حقوق به تفکیک زن و مرد

➤ فروشگاه‌های را در نظر بگیرید که درصدی از فروش به فروشنده اختصاص می‌دهد. میزان درصد به صورت زیر محاسبه می‌گردد:

کالای درجه ۱: اگر میزان فروش کم‌تر یا مساوی ۱۰۰۰۰۰۰ باشد، ۶ درصد به فروشنده تخصیص می‌یابد. اگر میزان فروش ۱۰۰۰۰۰۰ تا ۲۰۰۰۰۰۰ باشد ۷٪ و گرنه ۱۰٪ به فروشنده اختصاص می‌یابد.

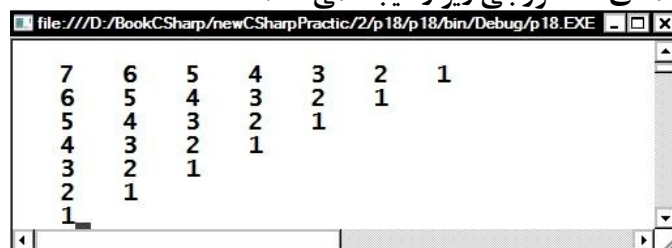
کالای درجه ۲: اگر میزان فروش کم‌تر از ۱۰۰۰۰۰۰ باشد، ۴٪ و گرنه ۶٪ به فروشنده اختصاص می‌یابد.

کالای درجه ۳: در این نوع کالا، ۴۱٪ میزان فروش به فروشنده تخصیص می‌یابد.

کالای درجه ۴: در این نوع کالا، ۵٪ میزان فروش به فروشنده اختصاص می‌یابد.

➤ برنامه‌ای که تعداد اقلام فروش را دریافت می‌کند. سپس به تعداد خوانده‌شده درجه کالا، قیمت واحد کالا، تعداد فروش را می‌خواند و مجموع فروش و مبلغی را چاپ می‌نماید که باید به فروشنده تخصیص یابد.

➤ برنامه‌ای که خروجی زیر را ایجاد می‌کند.



7	6	5	4	3	2	1
6	5	4	3	2	1	
5	4	3	2	1		
4	3	2	1			
3	2	1				
2	1					
1						

➤ بازی سنگ، کاغذ و قیچی دو بازیکن دارد. بازیکن اول می‌تواند قیچی (مقدار ۱)، سنگ (مقدار ۲) و یا کاغذ (مقدار ۳) را انتخاب کند و بازیکن دوم همین‌طور. اگر انتخاب دو بازیکن یکسان باشد، مساوی‌اند. اگر متفاوت باشد، یک نفر برنده است و برنده یک امتیاز را کسب می‌کند. برای مثال:

اگر بازیکن اول قیچی و بازیکن دوم کاغذ را انتخاب کند، بازیکن اول برنده است. چون قیچی کاغذ را می‌برد.

اگر بازیکن اول قیچی و بازیکن دوم سنگ را انتخاب کند، بازیکن دوم برنده است. زیرا سنگ قیچی را می‌شکند.

اگر بازیکن اول سنگ و بازیکن دوم کاغذ را انتخاب کند، بازیکن دوم برنده است. زیرا کاغذ دور سنگ می‌پیچد.

برنامه‌ای که این مسابقه را ۵ دور بین دو بازیکن انجام می‌دهد و برنده را نمایش می‌دهد (بازیکن‌ها از انتخاب هم دیگر خبر ندارند).

➤ برنامه‌ای که تمام اعداد بین ۱۰۰۰ تا ۱۱۰۰ که بر ۹ بخش پذیرند را نمایش می‌دهد (عددی بر ۹ بخش پذیر است که مجموع ارقام آن بر ۹ قابل قسمت باشد).

## ۶۰۰ برنامه C# با حل آن‌ها ۴۵

➤ برنامه‌ای که کاراکتری را که نشان‌دهنده رنگی می‌باشد، از ورودی خوانده، می‌گوید چه رنگی را انتخاب کردید. مثلاً اگر 'r' یا R را وارد کردید، برنامه به شما می‌گوید دوست دارید رنگ قرمز (Red) را انتخاب کنید. برای خروج کاربر باید e یا E را وارد کنید.

➤ برنامه‌ای که دو عدد مثبت را از ورودی خوانده، آنگاه عدد بزرگ‌تر را به روش تفریق بر عدد کوچک‌تر تقسیم می‌کند.

➤ برنامه‌ای که n را می‌خواند و حاصل عبارت زیر را محاسبه می‌کند:

$$\frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$$

➤ برنامه‌ای که با استفاده از حلقه تودرتو خروجی زیر را ایجاد می‌کند:

```
file:///D:/BookCSharp/newCSharpPractic/2/p24/p24/bin/Debug/p24.EXE
$$$$$$
$$$$$$
$$$$$$
$$$$$$
```

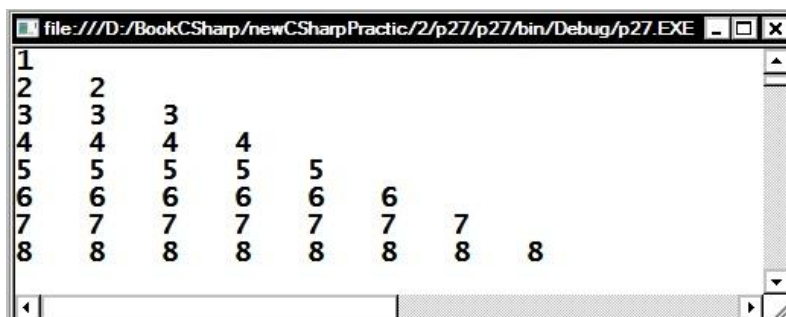
➤ برنامه‌ای که با استفاده از حلقه تودرتو خروجی زیر را چاپ می‌کند:

```
file:///D:/BookCSharp/newCSharpPractic/2/p25/p25/bin/Debug/p25.EXE
1 1 1 1 1 1 1 1
2 2 2 2 2 2 2
3 3 3 3 3 3
4 4 4 4 4
5 5 5 5
6 6 6
7 7
8
```

➤ برنامه‌ای که با استفاده از حلقه تودرتو خروجی زیر را چاپ می‌کند:

```
file:///D:/BookCSharp/newCSharpPractic/2/p26/p26/bin/Debug/p26.EXE
1 1 1 1 1 1 1 1
2 2 2 2 2 2 2
3 3 3 3 3 3 3
4 4 4 4 4 4 4
5 5 5 5 5 5 5
6 6 6 6 6 6 6
7 7 7 7 7 7 7
8 8 8 8 8 8 8
```

➤ برنامه‌ای که با استفاده از حلقه تودرتو خروجی زیر را نمایش می‌دهد:



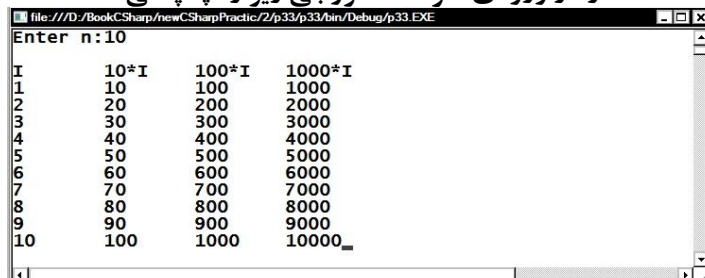
- برنامه‌ای که القاب افراد را از ورودی به صورت یک کاراکتر خوانده، سپس لقب آن را به صورت کامل در خروجی چاپ می‌کند (بر اساس جدول زیر):

علامت ورودی	لقب	لقب لاتین
b, B	بانو	lady
d, D	دوشیزه	miss
p, P	پروفسور	professor
a, A	آقا	mr
j, J	جناب	excellency
m, M	خانم	wife

- قیمت اتومبیل در انتهای هر سال ۲۰٪ کاهش می‌یابد (استهلاک اتومبیل ۲۰٪ است).  
برنامه‌ای که قیمت یک اتومبیل را از ورودی خوانده، تا ده سال آینده قیمت آن را در پایان هر سال نمایش می‌دهد.
- برنامه‌ای که عددی را از ورودی خوانده، تشخیص می‌دهد که این عدد جزء سری فیبوناچی است یا خیر؟
- برنامه‌ای که تمام حالت‌هایی که یک اسکناس ۱۰۰۰ ریالی را با ۵۰۰، ۲۰۰، ۱۰۰ و ۵۰ ریالی می‌توان خرد نمود می‌نویسد.
- برای رانندگان میزان مصرف سوخت خودرویشان از اهمیت ویژه‌ای برخوردار است. راننده‌ای چند بار مقدار بنزین باک خودروی خود را به همراه مقدار مسافتی که رانندگی کرده، ثبت نموده است. برنامه‌ای که مسافت را به مایل و مصرف بنزین را به گالن از کاربر می‌گیرد و نسبت مایل به گالن را برای هر بار محاسبه می‌کند و نمایش می‌دهد. برنامه پس از پردازش همه اطلاعات نسبت مایل به گالن کل را محاسبه نموده، چاپ می‌کند. برای خروج از برنامه به جای گالن مقدار ۰ را وارد کنید.

## ۶۰۰ برنامه C# با حل آن‌ها ۴۷

➤ برنامه‌ای که عدد N را از ورودی خوانده، خروجی زیر را چاپ می‌کند:



I	10*I	100*I	1000*I
1	10	100	1000
2	20	200	2000
3	30	300	3000
4	40	400	4000
5	50	500	5000
6	60	600	6000
7	70	700	7000
8	80	800	8000
9	90	900	9000
10	100	1000	10000

➤ برنامه‌ای که دنباله‌ای از اعداد صحیح را باهم جمع می‌کند. فرض کنید که اولین عدد صحیحی که خوانده می‌شود، تعداد اعدادی را که باید وارد شوند، مشخص می‌کند. به عنوان مثال، ورودی به صورت زیر می‌باشد:

➤ ۸۰۰ ۲۷۰ ۲۰۰ ۱۷۰ ۱۵۰ ۱۰۰ ۶

➤ برنامه میانگین و مجموع اعداد ۱۰۰، ۱۵۰، ۱۷۰، ۲۰۰، ۲۷۰ و ۸۰۰ را نمایش می‌دهد.

➤ برنامه‌ای که مقدار سری  $\pi$  (سری زیر) را حداکثر تا n جمله محاسبه می‌کند و نمایش می‌دهد (n را از ورودی می‌خواند):

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$$

➤ برنامه جدولی چاپ می‌کند که حاصل مقدار  $\pi$  را با تقریب یک جمله، دو جمله، سه جمله، ...، n جمله نمایش می‌دهد.

➤ برنامه‌ای که سه ضلع (عدد صحیح غیر صفر) را می‌خواند و تشخیص می‌دهد آیا سه ضلع اضلاع یک مثلث هستند یا خیر. (توضیح: سه ضلع زمانی تشکیل مثلث را می‌دهند که مجموع هر دو ضلع بیش‌تر از ضلع سوم باشد.)

➤ برنامه‌ای که سه عدد صحیح غیر صفر را گرفته، تشخیص دهد این سه عدد تشکیل مثلث قائم‌الزاویه را می‌دهند یا خیر؟ زمانی که توان ۲ یکی از اضلاع برابر مجموع توان ۲ اضلاع دیگر باشد مثلث قائم‌الزاویه است.  
(توضیح: هدف از این برنامه کاربرد عملگر || (یا منطقی) می‌باشد.)

➤ نمره دانشجویان یک کلاس به صورت زیر محاسبه می‌شود:  
در طول ترم دو کوئیز برگزار می‌گردد که هر کدام ۱۰ نمره دارند.  
امتحان پایان ترم ۵۰ درصد نمره کل، امتحان میان ترم ۲۵ درصد نمره کل و ۲۵ درصد دیگر مربوط به کوئیزها است.

برنامه‌ای که نمرات کوئیز میان ترم و پایان ترم یک دانشجو را گرفته، نمره کل او را محاسبه می‌کند و به صورت حروفی نمایش می‌دهد (برای نمرات بزرگ‌تر یا مساوی ۹۰، حروف A، نمرات بزرگ‌تر یا مساوی ۸۰ و کم‌تر از ۹۰ حرف B، نمرات بزرگ‌تر یا مساوی ۷۰ و کم‌تر از

۸۰ حرف C، نمرات بزرگ‌تر یا مساوی ۶۰ و کم‌تر از ۶۰، حرف D و نمرات زیر ۶۰ حرف E را نمایش دهد). (توضیح: در این برنامه، ابتدا نمرات کوئیز را به ۱۰۰ تبدیل می‌کنیم. یعنی، نمره کوئیز اول + نمره کوئیز دوم را در ۵ ضرب می‌کنیم. سپس برای محاسبه نمره کل، نمره به‌دست آمده از کوئیز را در ۰,۲۵ ضرب کرده، با ۰,۲۵ نمره میان‌ترم جمع نمود. حاصل را با ۰,۵ پایان‌ترم جمع می‌نماییم تا نمره کل به ۱۰۰ به دست آید).

➤ برنامه‌ای که تعیین می‌کند در یک فروشگاه آیا میزان خرید مشتری از سقف اعتبار حسابش بیش‌تر شده است یا خیر. برای هر مشتری اطلاعات زیر موجود می‌باشد:

۱. شماره حساب (عدد صحیح) ۲. تراز اول ماه

۳. مجموع همه اقلامی که در یک ماه توسط این مشتری خریداری شده است.

۴. مجموع همه اعتباراتی که در یک ماه به حساب مشتری وارد شده است.

۵. سقف اعتبار مجاز

این برنامه باید هر یک از اطلاعات ۱ تا ۵ را بگیرد و تراز جدید (= اعتبارات - هزینه‌ها + تراز اولیه) را محاسبه و تعیین کند که آیا تراز جدید از سقف اعتباراتشان بیش‌تر شده است. برنامه باید شماره حساب، سقف اعتبارات و تراز مشتری را به همراه پیام 'Credit Limited Exceeded' (از سقف اعتبارات تجاوز کرده است) چاپ کند.

➤ برنامه‌ای که n را خوانده، علاوه بر نمایش عبارت زیر، مجموع عبارت را نیز نمایش می‌دهد:

$$مجموع = 1 + 2 + 3 + \dots + n$$

➤ برنامه‌ای که n را خوانده، علاوه بر نمایش عبارت زیر، حاصل عبارت را نیز نمایش می‌دهد:

$$حاصل عبارت = 1 - 2 + 3 - 4 + \dots \pm n$$

➤ برنامه‌ای که n را خوانده، علاوه بر نمایش عبارت زیر، حاصل عبارت را نیز نمایش می‌دهد:

$$حاصل عبارت = -1 + 2 - 3 + 4 + \dots \pm n$$

➤ برنامه‌ای که n را خوانده، علاوه بر نمایش عبارت زیر، حاصل عبارت را نیز نمایش می‌دهد:

$$حاصل عبارت = 1 - \frac{2}{2!} + \frac{3}{3!} - \frac{4}{4!} + \dots \pm \frac{n}{n!}$$

➤ برنامه‌ای که x و n را خوانده، علاوه بر نمایش عبارت زیر، حاصل عبارت را نیز نمایش می‌دهد:

$$حاصل عبارت = x - \frac{x^2}{2!} + \frac{x^3}{3!} - \frac{x^4}{4!} + \dots \pm \frac{x^n}{n!}$$

## ۶۰۰ برنامه C# با حل آن‌ها ۴۹

- برنامه‌ای که x و n را خوانده، علاوه بر نمایش عبارت زیر، حاصل عبارت را نیز نمایش می‌دهد:

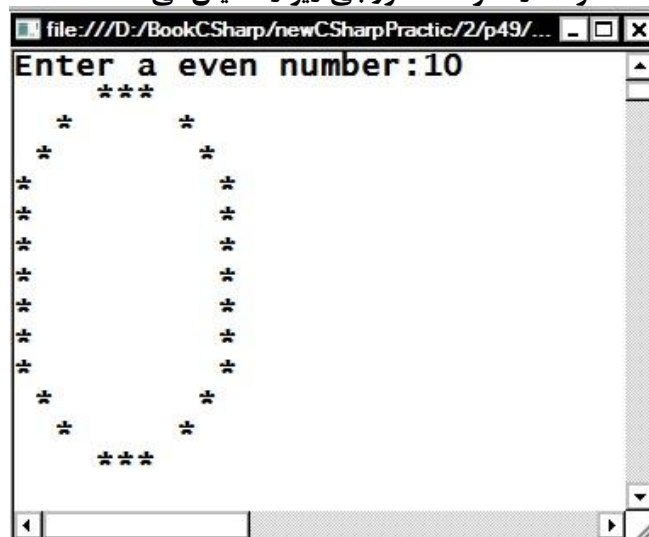
$$\text{حاصل عبارت} = \frac{x^2}{2!} - \frac{x^4}{4!} + \frac{x^6}{6!} - \dots \pm \frac{x^n}{n!}$$

- برنامه‌ای که یک عدد، یک کاراکتر و سپس عدد دیگری را خوانده، اعمال زیر را انجام دهد:

اگر کاراکتر ورودی "A"، "a" یا "+" باشد، حاصل جمع دو عدد را نمایش دهد.  
 اگر کاراکتر ورودی "S"، "s" یا "-" باشد، حاصل تفریق دو عدد را نمایش دهد.  
 اگر کاراکتر ورودی "X"، "x" یا "\*" باشد، حاصل ضرب دو عدد را نمایش دهد.  
 اگر کاراکتر ورودی "D"، "d" یا "/" باشد، حاصل تقسیم دو عدد را نمایش دهد.  
 اگر کاراکتر ورودی "M"، "m" یا "%" باشد، باقی‌مانده تقسیم صحیح عدد اول بر عدد دوم را نمایش دهد.  
 در غیر این صورت، پیغام "Invalid operator" را نمایش دهد.

- برنامه قبلی را با دستور switch پیاده‌سازی می‌کند.
- برنامه‌ای که دو عدد را خوانده، اگر باقی‌مانده تقسیم هر عدد بر عدد دیگر صفر شود، "Yes"، وگرنه "No" را نمایش دهد.

- برنامه‌ای که عدد فرد n را خوانده، خروجی زیر را نمایش می‌دهد :



➤ برنامه‌ای که کاراکتر به همراه کد اسکی تمام کاراکترهای رایانه را نمایش می‌دهد (متغیر i شمارنده‌ای که صفر تا ۲۵۵ را می‌شمارد. کد اسکی کاراکترهای رایانه از صفر تا ۲۵۵ می‌باشد).

➤ برنامه‌ای که خروجی زیر را تولید کرده، نمایش می‌دهد.

1									
2	4								
3	6	9							
4	8	12	16						
5	10	15	20	25					
6	12	18	24	30	36				
7	14	21	28	35	42	49			
8	16	24	32	40	48	56	64		
9	18	27	36	45	54	63	72	81	
10	20	30	40	50	60	70	80	90	100

➤ برنامه‌ای که تعدادی عددی را دریافت کرده، معادل ماه آن‌ها را نمایش می‌دهد (اگر عدد وارد شده خارج از بازه ۱ تا ۱۲ باشد، برنامه خاتمه می‌یابد). در این برنامه month عددی است که ماه معادل آن باید تعیین شود.

➤ برنامه‌ای که رشته‌ای را دریافت کرده، تعداد کاراکترهای رشته را می‌شمارد (رشته به Enter ختم می‌شود).

➤ برنامه‌ای که حقوق تعدادی کارمند را خوانده، اگر کارمند بیش از ۴۰ ساعت کار کرده باشد، به اندازه ۱,۵ برابر هر ساعت به او اضافه کاری داده می‌شود. در پایان، کل حقوق دریافتی را نمایش می‌دهد (برای خروج از برنامه از کاربر سؤال می‌کند که آیا ادامه می‌دهد یا خیر؟ اگر کاربر صفر را وارد کند، برنامه خاتمه می‌یابد).

➤ برنامه‌ای که نتایج عملگرهای (AND)&&, (OR)|, (NOT)! را به عدد تبدیل کرده، نمایش دهد.

➤ برنامه‌ای که عددی را خوانده، در هر مرحله عدد موردنظر، رقم یکان و حاصل ضرب ارقام فرد آن را نمایش می‌دهد. به عنوان مثال، اگر کاربر عدد ۳۲۷۶ را وارد نماید خروجی به صورت زیر نمایش داده می‌شود:

Number	Digit	Multiply
3276	6	1
327	7	7
32	2	7
3	3	21
Multiply		is 21

## ۶۰۰ برنامه C# با حل آن ها ۵۱

➤ برنامه‌ای که عددی را خوانده، میانگین ارقام فرد و زوج آن را نمایش می‌دهد. به عنوان مثال، اگر عدد ۲۵۶۹ باشد، میانگین ارقام فرد آن، یعنی  $(۵+۹)/۲$  (۷) است. میانگین ارقام زوج آن  $(۲+۶)/۲$  (۴) می‌باشد.

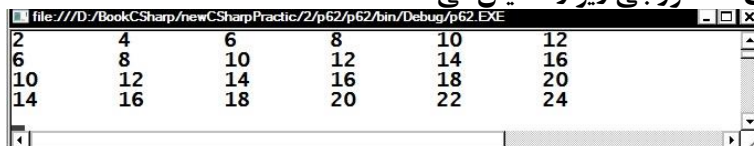
➤ برنامه‌ای که کلیه اعداد سه‌رقمی را نمایش می‌دهد که با ارقام ۱، ۲ و ۳ تولید می‌شوند.

➤ برنامه‌ای که کلیه اعداد سه‌رقمی را نشان می‌دهد که در ارقام آن ۱، ۴ یا ۸ باشد.

➤ برنامه‌ای که کلیه اعداد چهاررقمی که فقط از ارقام ۱، ۲، ۴ و ۹ تشکیل شده باشند و هر رقم یک‌بار تکرار شده باشد را نمایش می‌دهد.

➤ برنامه‌ای که یک جمله را خوانده، تعداد ارقام ۰ تا ۹ آن را می‌شمارد (جمله با Ctrl + z یعنی EOF خاتمه می‌یابد).

➤ برنامه‌ای که خروجی زیر را نمایش می‌دهد:



2	4	6	8	10	12
6	8	10	12	14	16
10	12	14	16	18	20
14	16	18	20	22	24

➤ برنامه‌ای که خروجی زیر را نمایش می‌دهد:

➤ برنامه‌ای که x و n را خوانده، حاصل عبارت  $\sum_{i=1}^n x^n$  را نمایش می‌دهد (یعنی، حاصل عبارت زیر را نمایش می‌دهد):

$$x^1 + x^2 + x^3 + \dots + x^n$$

➤ برنامه‌ای که x، n و k را خوانده، حاصل عبارت  $F(x, n, k) = \frac{\sum_{j=n}^0 x^j}{\sum_{i=1}^k i!}$  را نمایش می‌دهد.

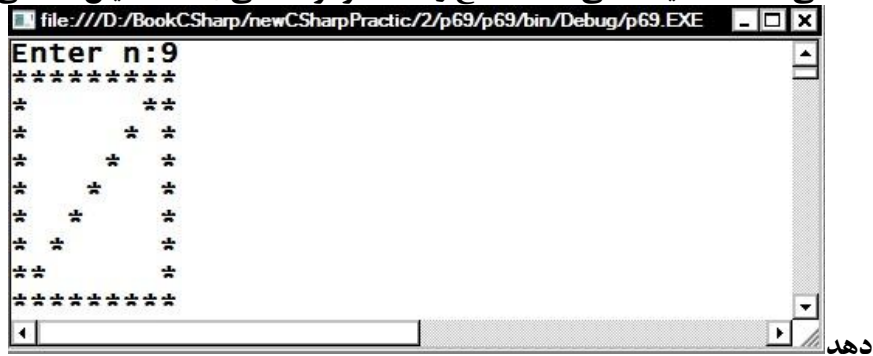
➤ برنامه‌ای که n (عددی بین ۱ تا ۸۰) را خوانده، خروجی زیر را نمایش می‌دهد.

➤ برنامه‌ای که n (عددی بین ۱ تا ۸۰) را خوانده، خروجی زیر را نمایش می‌دهد.

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مانند زیر را نمایش می‌دهد.

### ۶۰۰ برنامه C# با حل آن‌ها ۵۳

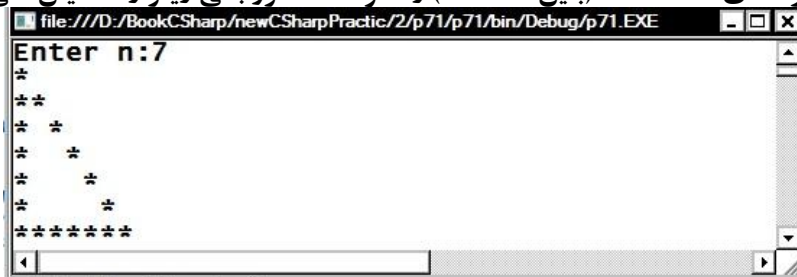
➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را از ورودی خوانده، خروجی مثل زیر را نمایش می‌دهد. یعنی، اضلاع و قطر فرعی را نمایش می‌دهد.



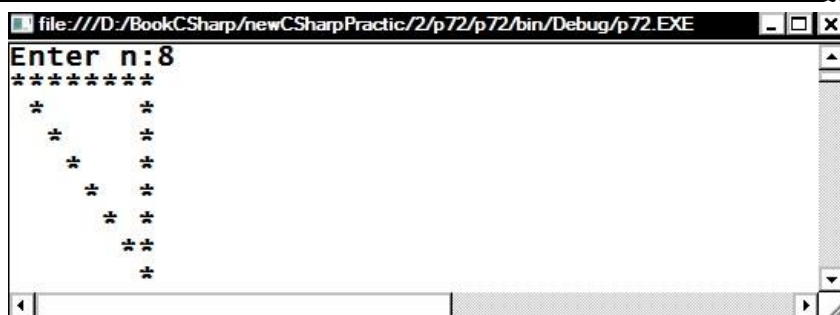
➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مثل زیر را نمایش می‌دهد.



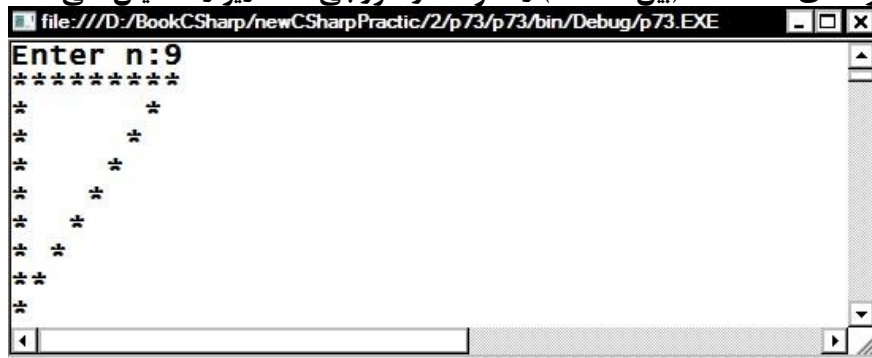
➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مثل زیر را نمایش می‌دهد.



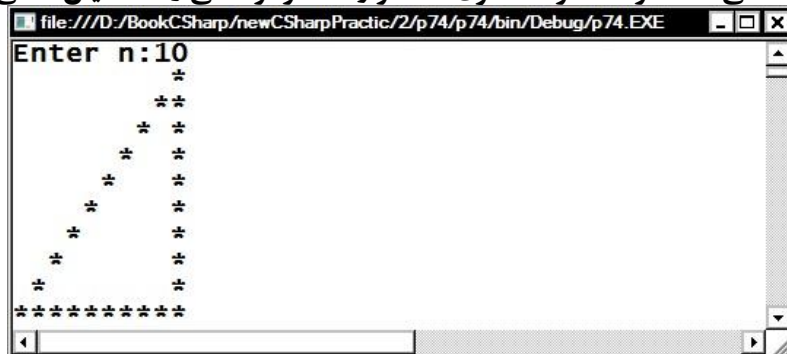
➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مانند شکل زیر را نمایش می‌دهد.



➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده و خروجی مانند زیر را نمایش می‌دهد.



➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مثل زیر را نمایش می‌دهد. یعنی، سطر آخر، ستون آخر و قطر فرعی را نمایش می‌دهد.

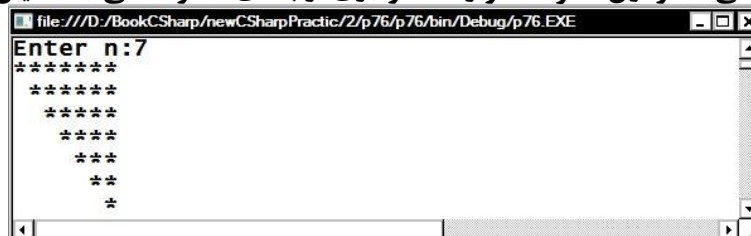


➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مثل زیر را نمایش می‌دهد. یعنی، سطر آخر، ستون اول و تمام عناصر روی قطر اصلی و زیر آن را نمایش می‌دهد.



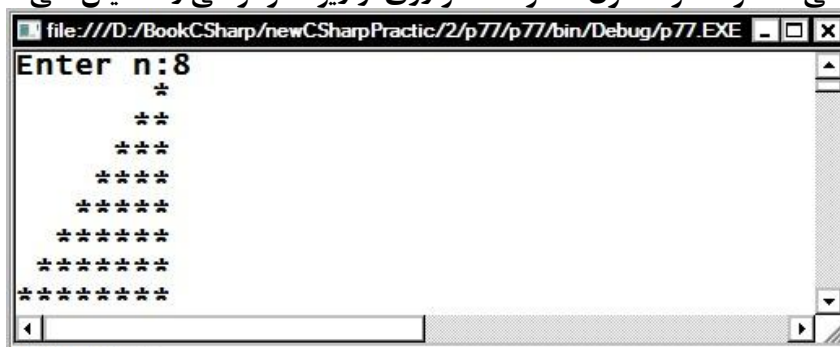
## ۶۰۰ برنامه C# با حل آن‌ها ۵۵

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مثل زیر را نمایش می‌دهد. یعنی، سطر اول، ستون آخر و عناصر روی و بالای قطر اصلی را نمایش می‌دهد.



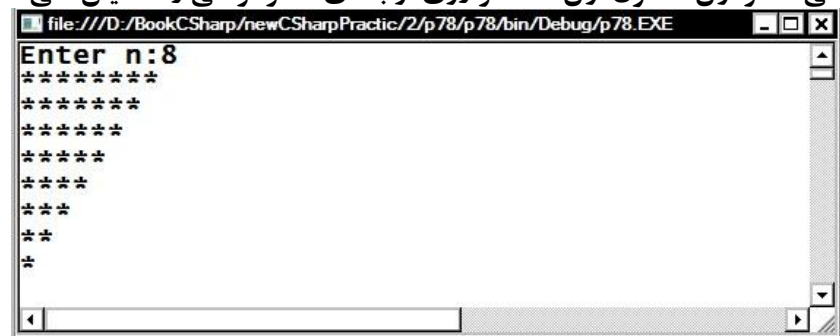
```
file:///D:/BookCSharp/newCSharpPractic/2/p76/p76/bin/Debug/p76.EXE
Enter n:7
*****
*****
*****
****
***
**
*
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مثل زیر را نمایش می‌دهد. یعنی، سطر آخر، ستون آخر، عناصر روی و زیر قطر فرعی را نمایش می‌دهد.



```
file:///D:/BookCSharp/newCSharpPractic/2/p77/p77/bin/Debug/p77.EXE
Enter n:8
*
**
***
****
*****
*****
*****
*****
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مثل زیر را نمایش می‌دهد. یعنی، سطر اول، ستون اول، عناصر روی و بالای قطر فرعی را نمایش می‌دهد.



```
file:///D:/BookCSharp/newCSharpPractic/2/p78/p78/bin/Debug/p78.EXE
Enter n:8
*****
*****
*****
*****
****
***
**
*
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مثل زیر را نمایش می‌دهد.

```

Enter n:8
*
**
***
****
*****
****
***
**
*
  
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی زیر را نمایش می‌دهد.

```

Enter n:9
*****
*
*
*
*
*
*
*
*
*
  
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی زیر را نمایش می‌دهد.

```

Enter n:10
*
**
***
****
*****
*****
*****
****
***
**
*
  
```

## ۶۰۰ برنامه C# با حل آن‌ها ۵۷

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی مثل زیر را نمایش می‌دهد.

```
file:///D:/BookCSharp/newCSharpPractic/2/p82/p82/bin/Debug/p82.EXE
Enter n:10
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی زیر را نمایش می‌دهد.

```
file:///D:/BookCSharp/newCSharpPractic/2/p83/p83/bin/Debug/p83.EXE
Enter n:8
*
**
***
****
*****
*****
*****
*****
*
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی زیر را نمایش می‌دهد.

```
file:///D:/BookCSharp/newCSharpPractic/2/p84/p84/bin/Debug/p84.EXE
Enter n:10
*
**
***
****
*****
*****
*****
*****
*****
*
```

➤ برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی زیر را نمایش می‌دهد.

```
file:///D:/BookCSharp/newCSharpPractic/2/p85/p85/bin/Debug/p85.EXE
Enter n:9
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

- برنامه‌ای که عدد n (بین ۱ تا ۸۰) را خوانده، خروجی زیر را نمایش می‌دهد.



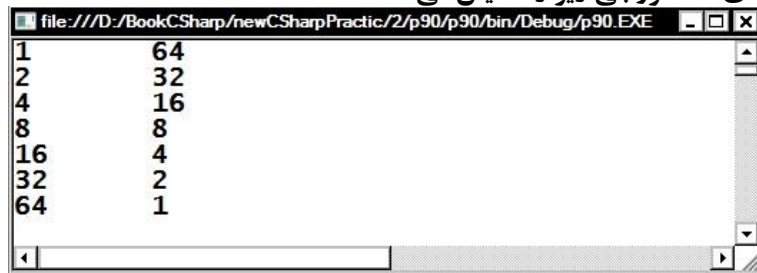
- برنامه‌ای که حقوق کارمند، نوع کسور بیمه و درصد مالیات را خوانده، حقوق خالص کارمند را حساب می‌نماید. برای محاسبه حقوق خالص به صورت زیر عمل می‌گردد (برای حل این تمرین از عملگر ؟ استفاده کنید):

اگر نوع کسور بیمه ۱ باشد،  $۷ / ۱۰۰ * \text{حقوق} = \text{بیمه}$   
 اگر نوع کسور بیمه ۲ باشد،  $۸,۵ / ۱۰۰ * \text{حقوق} = \text{بیمه}$   
 اگر نوع کسور بیمه ۳ باشد،  $۱۰ / ۱۰۰ * \text{حقوق} = \text{بیمه}$   
 $۱۰۰ / \text{درصد مالیات} * \text{حقوق} = \text{مالیات}$   
 $\text{مالیات} - \text{بیمه} - \text{حقوق} = \text{حقوق خالص}$

- برنامه‌ای که دو عدد صحیح x و y را خوانده، با استفاده از عملگر ضرب،  $x^y$  را محاسبه می‌کند و نمایش می‌دهد:

- برنامه‌ای که تعداد عدد را خوانده و تشخیص می‌دهد هر عدد خوانده شده دارای رقم صفر است یا خیر. برای خروج از برنامه کاربر باید ۱- را وارد کند.

- برنامه‌ای که خروجی زیر را نمایش می‌دهد:



- برنامه‌ای که شماره خانوار، هزینه روزانه آن‌ها را به روز می‌خواند، هزینه ماهیانه و سالیانه آن‌ها را نمایش می‌دهد. سپس تعیین می‌کند کدام خانوار کم‌ترین هزینه و کدام خانوار بیش‌ترین هزینه را دارد. برای خروج از برنامه به جای شماره خانوار عدد ۹۹- وارد می‌شود.

- برنامه‌ای که عدد n را خوانده (n تعداد کارمندان است). سپس ساعات اضافه کار n کارمند را خوانده و سه کارمند که کم‌ترین اضافه کار را دارند، نمایش می‌دهد.
- برنامه‌ای که سن و جنسیت تعدادی افراد را خوانده، تعداد و درصد مردانی که سن آن‌ها بین ۱۸ تا ۲۰ است را محاسبه می‌کند و نمایش می‌دهد. اگر جنسیت M یا m وارد گردید، این فرد، مرد است (برای خاتمه برنامه به جای سن ۱- وارد می‌شود).
- یک شرکت بیمه به بیمه‌گذاران خود سود سالانه می‌دهد. برنامه‌ای که میزان سود سالانه (درصد سود سالانه) و میزان سرمایه‌گذاری یک بیمه‌گذار و تعداد سال‌های سرمایه‌گذاری را دریافت می‌کند و سود بیمه‌گذار را برای n سال مختلف حساب می‌نماید (بدون پرداخت سود به بیمه‌گذار یعنی سود به سرمایه سالانه اضافه گردد). سرمایه بیمه‌گذار پس از n سال چقدر است.
- برنامه‌ای که تمام اعداد چهاررقمی را نمایش می‌دهد که حاصل ضرب ارقام آن‌ها عدد تام (کامل) است. عددی تام است که مجموع مضرب‌های اعداد کوچک‌تر از خودش برابر خودش باشد. به‌عنوان مثال، دو عدد زیر تام هستند:  

$$۶ = ۱ + ۲ + ۳$$

$$۲۸ = ۱ + ۲ + ۴ + ۷ + ۱۴$$
- برنامه‌ای که تعداد رقم‌های صفر اعداد چهاررقمی را می‌شمارد.
- برنامه‌ای که سه عدد را خوانده، اعداد بین اعداد اول و دوم که مجموع ارقام آن‌ها برابر عدد سوم باشد را نمایش می‌دهد. به‌عنوان مثال، اگر کاربر ۵، ۱۰۰۰ و ۱۵ را وارد کند. برنامه باید اعداد از ۵ تا ۱۰۰۰ که مجموع ارقام آن‌ها برابر ۱۵ باشد را نمایش دهد.
- برنامه‌ای که دو عدد را خوانده، اعداد اولی که بین این دو عدد قرار دارند را نمایش می‌دهد (عددی اول است که بر هیچ عدد کوچک‌تر از نصف خودش به جزء یک بخش پذیر نباشد).
- برنامه‌ای که یک عدد صحیح را خوانده و ارقامی از آن عدد که بر ۲ یا ۳ بخش پذیر هستند را نمایش می‌دهد.
- هنگام خرید خانه عوامل مختلفی را باید در نظر گرفت. در هنگام خرید خانه، میزان هزینه تقریبی سوختی که برای یک سال مصرف می‌شود و نرخ مالیات سالانه داده‌شده‌اند. برنامه‌ای که داده‌هایی از قبیل قیمت خانه، هزینه سوخت سالانه و مالیات

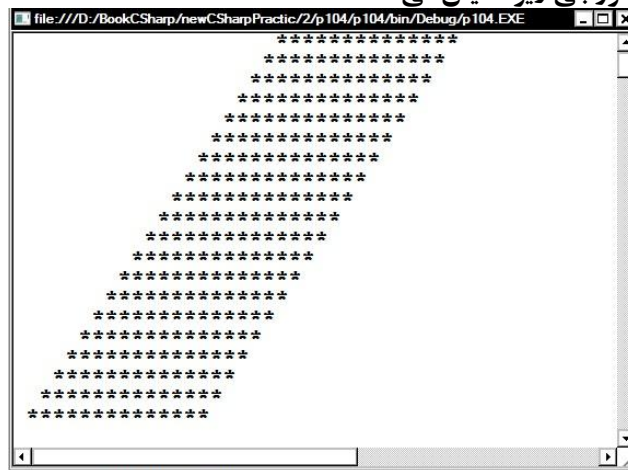
سالانه را دریافت می کند. این برنامه هزینه ۱۰ ساله خانه را محاسبه می نماید. برنامه تا زمانی که کاربر به جای قیمت خانه ۹۹۹- وارد کند، ادامه می یابد.

➤ برنامه ای که عدد  $n$  را خوانده، مقدار  $\sqrt{n + \sqrt{n + \sqrt{n + \dots}}}$  را تا دقت  $10^{-10}$  محاسبه می کند و نمایش می دهد.

➤ در یک بازی دونفره  $n$  چوب کبریت وجود دارد. هر یک از بازیکنان می توانند در نوبت خود، ۱، ۲ یا حداکثر ۳ چوب کبریت را بردارند. بازیکنی که آخرین چوب کبریت را برمی دارد، بازنده است. برنامه ای که این بازی را بین دو بازیکن اجرا می کند. این برنامه، ابتدا تعداد چوب کبریت ها را خوانده و در هر مرحله تعداد چوب کبریت های باقی مانده را نمایش می دهد.

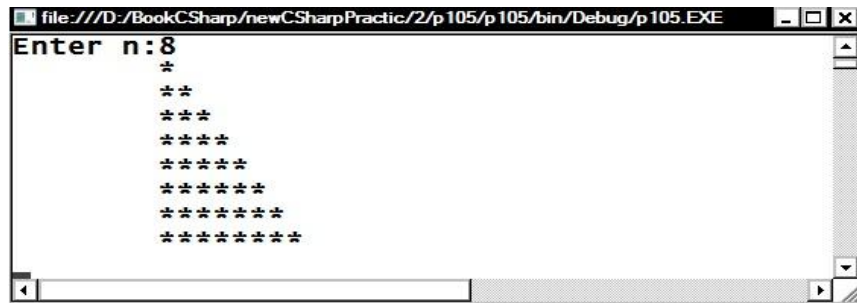
➤ برنامه ای که اگر کاربر هر یک از حروف A, B, C و D را وارد کند، پیام Big Character! را نمایش می دهد. اگر کاربر یکی از حروف a, b, c یا d را وارد کند، Litter character! را نمایش خواهد داد. ولی اگر کاربر یکی از کاراکترهای ۱، ۲، ۳ یا ۴ را وارد نماید، پیام A Figurer! را نمایش می دهد. وگرنه پیام Unknown Character! را نمایش می دهد (هدف این برنامه or کردن مقادیر در دستور switch است).

➤ برنامه ای که خروجی زیر نمایش می دهد:



➤ برنامه ای که  $n$  را خوانده، خروجی مانند زیر نمایش را می دهد:

## ۶۰۰ برنامه C# با حل آن‌ها ۶۱



➤ ۱۰۶. برنامه‌ای که سه ضلع مثلث را خوانده و مشخص می‌کند که آیا مثلث متساوی‌الاضلاع P، متساوی‌الساقین است یا مختلف‌الاضلاع می‌باشد (در این برنامه متغیرهای a، b و c اضلاع مثلث را نگهداری می‌کنند).

➤ ۱۰۷. برنامه‌ای که سه ضلع مثلث را دریافت کرده، اگر مثلث قائم‌الزاویه بود، مساحت آن، وگرنه محیط آن را محاسبه کرده و نمایش می‌دهد. در این برنامه متغیرهای a، b و c اضلاع مثلث را نگهداری می‌کنند، اما متغیر temp کمکی است که محیط یا مساحت را ذخیره می‌نماید.

➤ ۱۰۸. برنامه‌ای که کاراکتری را خوانده و مشخص می‌کند که در کدام یک از دسته‌های زیر قرار دارد؟  
☐ حروف بی‌صدا    ☐ حروف صدادار    ☐ ارقام    ☐ سایر کاراکترها

➤ ۱۰۹. برنامه‌ای که دو عدد صحیح را خوانده و مقسوم‌علیه مشترکشان را نمایش می‌دهد. در این برنامه x و y دو عدد هستند و i شمارنده از ۱ تا x می‌باشد.

➤ ۱۱۰. برنامه‌ای که ابتدا n را خوانده، سپس n معدل را برای n دانشجو می‌خواند و بزرگ‌ترین معدل مشروط را نمایش می‌دهد. در این برنامه n، تعداد دانشجویان، grade، معدل هر دانشجو و max بزرگ‌ترین معدل مشروط شده است.

➤ ۱۱۱. برنامه‌ای که معدل n دانشجو را خوانده، میانگین معدل‌های بالاتر از ۱۵ را نمایش می‌دهد. در این برنامه n (تعداد دانشجویان)، grade (معدل هر دانشجو)، sum (مجموع معدل‌های بالای ۱۵)، count (تعداد معدل‌های بالای ۱۵) و i (شمارنده) است.

➤ ۱۱۲. برنامه‌ای که حاصل سری زیر را تا ۴ رقم اعشار محاسبه کرده، نمایش می‌دهد:

$$+\frac{1}{2}-\left(\frac{1}{2}\times\frac{1}{3}\right)+\left(\frac{1}{2}\times\frac{1}{3}\times\frac{1}{4}\right)-\dots$$

➤ ۱۱۳. برنامه‌ای که ابتدا تعداد دانشجویان ( $n$ ) را خوانده، سپس  $n$  معدل را می‌خواند و تعداد دانشجویایی که معدل آن‌ها بین ۱۰ تا ۱۵ است را شمارش می‌کند و نمایش می‌دهد. در این برنامه  $n$  (تعداد دانشجویان)،  $grade$  (معدل هر دانشجو)،  $i$  (شمارنده) و  $count$  (تعداد دانشجویانی که معدل آن‌ها بین ۱۰ تا ۱۵) است.

➤ ۱۱۴. برنامه‌ای که ابتدا  $n$  را خوانده، سپس  $n$  جمله سری زیر را نمایش می‌دهد. در این سری زیر جمله‌ی سوم به بعد هر جمله میانگین دو جمله قبلی می‌باشد.  
 $1, 5, 3, 4, 3.5, \dots$

➤ ۱۱۵. برنامه‌ای که ابتدا  $n$  را خوانده، سپس  $n$  عدد را می‌خواند و واریانس آن را نمایش می‌دهد.

➤ پروژه برنامه‌نویسی ۱: برنامه‌ای که روزی از سال را خوانده، مشخص می‌کند که چه فصلی، چه ماه و چه روزی از سال است. به عنوان مثال، روز ۳۵ سال، فصل بهار، ماه اردیبهشت و روز چهارم این ماه است.

➤ پروژه برنامه‌نویسی ۲: برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل سری زیر را محاسبه می‌کند.

$$\frac{x^n n^x (x+n)^{n+x}}{n! x! (n+x)!} \sum_{i=1}^n \frac{x^i i^x}{n! i! (n+i)!}$$

➤ پروژه برنامه‌نویسی ۳: شرکت مخابرات ایران برای مکالمات راه دور از نرخ‌های زیر استفاده می‌کند:

تمام مکالماتی که بین ساعات ۲۳ شب تا ۸ صبح انجام می‌شود، نرخ مکالمات ۵۰٪ محاسبه می‌گردد.

برای مکالماتی که در روزهای تعطیل (روز ۷) انجام می‌شود، نرخ مکالمه ۷۵ درصد محاسبه می‌شود.

برای بقیه ساعات روزهای دیگر مکالمه به طور کامل حساب می‌شود.

نرخ هر پالس ۴۶ ریال می‌باشد.

به کلیه مکالمات ۴ درصد مالیات تعلق می‌گیرد.

برنامه‌ای که برای تعدادی مشترک، ساعت شروع مکالمه (از ساعت صفر تا ۲۳)، تعداد پالس-های مکالمه، روز انجام مکالمه را دریافت می‌کند، مبلغ ناخالص مکالمه، مالیات، مبلغ خالص، مجموع مالیات و مجموع مبلغ خاص را محاسبه و نمایش می‌دهد (برای خاتمه کاربر به جای ساعت شروع ۹۹- را وارد نماید).

➤ پروژه برنامه نویسی ۴: نرخ هر متر مکعب آب با توجه به نرخ مصرف از قبیل مصارف خانگی، تجاری یا صنعتی فرق می‌کند. برنامه‌ای که میزان مصرف آب به متر مکعب و نوع مصرف تعدادی مشترک را خوانده، مبلغ صورت حساب آب آن‌ها را محاسبه و چاپ می‌کند. برای محاسبه مصرف صورت حساب آب به روش زیر عمل می‌نماید:

اگر نوع مصرف حرف H یا h باشد (مصرف خانگی)، به ازای هر ۱۰۰ متر مکعب مصرف ۵۰۰ ریال دریافت می‌شود.

اگر نوع مصرف حرف I یا i وارد شود (مصرف صنعتی تا چهار میلیون متر مکعب)، به ازای هر ۱۰۰۰ متر مکعب آب ۷۵۰ ریال و به ازای هر متر مکعب بیش از آن ۰/۰۰۰۲۵ به مبلغ قبض اضافه می‌گردد.

اگر نوع مصرف حرف E یا e وارد شود (مصرف تجاری)، در صورتی که مصرف آب تا دو میلیون متر مکعب باشد، به ازای هر ۱۵۰۰ متر مکعب مصرف ۶۰۰ ریال و به ازای هر متر مکعب بیش از دو میلیون ۰/۰۰۰۰۴ به مبلغ صورت حساب اضافه خواهد شد.

برای خاتمه برنامه کاربر به جای مصرف آب، عدد منفی وارد می‌نماید.

➤ برنامه‌ای که اعداد سه‌رقمی را چاپ می‌کند که مجموع فاکتوریل ارقام آن اعداد برابر با خود عدد باشد. به‌عنوان مثال، اگر  $n_1$ ،  $n_2$  و  $n_3$  به ترتیب رقم یکان، دهگان و صدگان باشند، فرمول زیر برقرار باشد:

$$n_3 n_2 n_1 = n_1! + n_2! + n_3!$$

➤ برنامه‌ای که کلیه اعداد چهاررقمی چاپ را می‌کند که مجموع رقم اول به توان ۱ و رقم چهارم به توان ۴ برابر با مجموع رقم دوم به توان ۲ و رقم سوم به توان ۳ باشد. به‌عنوان مثال، داریم:

$$۲۱۴۱ = ۲^۴ + ۱^۱ = ۱^۳ + ۴^۲ = ۱۶ + ۱ = ۱ + ۱۶ = ۱۷$$

➤ برنامه‌ای که دو عدد را خوانده، به متدی ارسال کند. متد، تفاضل حاصل ضرب و حاصل تقسیم آن‌ها را محاسبه کرده، به برنامه ارسال کند. متدها را به‌صورت قالب‌های متد پیاده‌سازی کنید و دو بار آن را برای اعداد صحیح و اعشاری فراخوانی نمایید. این متد با استفاده از Template پیاده‌سازی شود.

➤ برنامه‌ای که عدد صحیح مثبتی را خوانده، مجموع ارقام آن را محاسبه می‌کند. اگر حاصل مجموع ارقام، عدد یک‌رقمی نبود، این روند را ادامه داده تا نهایتاً عدد یک‌رقمی شود (از یک متد بازگشتی استفاده می‌کند).

➤ برنامه‌ای که اضلاع مثلثی را خوانده، مساحت آن را با متدی محاسبه می‌کند و به خروجی می‌برد. اگر اضلاع مثلثی  $a$ ،  $b$  و  $c$  باشد، مساحت آن با فرمول مقابل محاسبه می‌شود.

$$p = (a + b + c) / ۲$$

$$\text{مساحت مثلث} = \sqrt{p(p-a)(p-b)(p-c)}$$

➤ برنامه‌ای که یک عدد بزرگ (حتی ۱۰۰۰ رقمی) را گرفته، تعیین می‌کند بر ۱۵ بخش پذیر است یا خیر. عددی بر ۱۵ بخش پذیر است که بر ۳ و ۵ بخش پذیر باشد. عددی بر ۳ بخش پذیر است که مجموع ارقام آن بر ۳ بخش پذیر باشد و عددی بر ۵ بخش پذیر است که رقم یکان آن صفر یا ۵ باشد (رقم یکان بر ۵ بخش پذیر است).

➤ برنامه‌ای که یک رقم را از ورودی خوانده، تمام اعداد صحیح بین ۱ و ۱۰۰ را چاپ می‌کند، به‌طوری که رقم دریافتی در اعداد ۱ تا ۱۰۰، مجذور و مکعب این اعداد وجود داشته باشد. به‌عنوان مثال، رقم (۱) را از ورودی بخوانیم، عدد ۱۳ یکی از اعدادی است

که رقم یک در آن شرط صدق می‌کند، زیرا، در ۱۳، ۱۶۹ و ۲۱۹۷ وجود دارد (برای تشخیص وجود رقم در عدد، مجذور و مکعب عدد از یک متد استفاده کنید).

➤ برنامه‌ای که سه عدد را از ورودی می‌خواند و توسط متدی میانگین آن‌ها را محاسبه می‌کند و نمایش می‌دهد.

➤ برنامه‌ای که یک عدد را از ورودی می‌خواند و توسط یک متد بازگشتی، شمارش معکوس از آن عدد به یک را انجام دهد (شماره‌ها را در خروجی چاپ می‌کند).

➤ متدی که دو آرگومان را می‌پذیرد و آرگومان اول را به توان آرگومان دوم می‌رساند. آرگومان اول یک مقدار double و آرگومان دوم یک مقدار صحیح مثبت یا منفی است. برنامه‌ای که از این متد استفاده می‌کند.

➤ برنامه‌ای که نمایش زمان را از حالت ۲۴ ساعت به ۱۲ ساعت تبدیل می‌کند. مثلاً باید ۱۴:۲۵ را به صورت ۲:۲۵pm تبدیل نماید. این برنامه طوری طراحی شده است که دارای سه متد زیر می‌باشد.

➤ قیمت سهام معمولاً به صورت کسری بیان می‌شود، مثلاً  $2\frac{7}{8}$  یا  $8\frac{1}{2}$ . برنامه‌ای که ارزش سهام را به صورت دو عدد صحیح و کسری از کاربر می‌گیرد. قسمت کسری را نیز به صورت دو عدد صحیح (عدد اول صورت و عدد دوم مخرج) دریافت می‌نماید (بر اساس تبدیل سهام از یک متد استفاده می‌کند که سه پارامتر را گرفته، ارزش سهام را به صورت double برمی‌گرداند).

➤ پارکینگی برای توقف تا سه ساعت، حداقل ۲ دلار و برای هر ساعت اضافه یا بخشی از ساعت اضافه بر ۳ ساعت ۰/۵ دلار می‌گیرد. حداکثر مبلغ قابل پرداخت برای هر دوره ۲۴ ساعتی ۱۰ دلار است. فرض کنید که هیچ خودرویی بیش از ۲۴ ساعت توقف نمی‌کند. برنامه‌ای که زمان استفاده از پارکینگ را برای مشتریان دریافت می‌کند و مبلغ پرداختی مشتری، مبلغ کل پرداختی را چاپ نماید. کاربر برای هر مشتری ساعات و دقیقه توقف را وارد می‌کند. متدی مبلغ پرداختی را محاسبه و برمی‌گرداند.

➤ برنامه‌ای که با استفاده از چند متد اعمال زیر را انجام دهید:  
تمام اعداد دورقمی که رقم اول و دوم آن‌ها برابر باشند را چاپ کند.  
تمام اعداد سه‌رقمی که رقم وسط آن‌ها صفر باشد را چاپ کند.  
تمام اعداد سه‌رقمی که مجموع رقم اول و دوم کوچک‌تر از رقم سوم باشد.  
تمام اعداد چهاررقمی که قرینه یکدیگرند (مثلاً ۲۳۳۲). یعنی رقم اول برابر رقم چهارم و رقم دوم برابر رقم سوم باشد.

- برنامه‌ای که با استفاده از یک متد بازگشتی بزرگ‌ترین مقسوم‌علیه مشترک دو عدد را محاسبه و چاپ می‌کند.
- برنامه‌ای که اولین روز سال را می‌گیرد و تقویم سالانه را نمایش می‌دهد.
- برنامه‌ای که یک عدد زوج را گرفته، تمام زوج‌های اولی (دو عدد فرد اولی که مجموع آن‌ها برابر با آن عدد زوج باشد) را نمایش می‌دهد. به‌عنوان مثال، اگر کاربر عدد ۲۲ را وارد کند، خروجی به‌صورت زیر نمایش داده شود.
- $$۲۲ = ۱۱ + ۱۱ \quad ۲۲ = ۱۷ + ۵ \quad ۲۲ = ۱۹ + ۳$$
- یونانیان باستان اعداد را به‌صورت هندسی دسته‌بندی می‌کردند. به‌عنوان مثال، آن‌ها یک عدد را مثلثی می‌نامیدند که می‌توانستند با آن تعدادی ریگ (به‌اندازه عدد ریگ)، در یک تقارن مثلثی بچینند. متدی به نام isTrain که یک عدد را گرفته تشخیص می‌دهد، عدد مثلثی است یا خیر. اگر عدد مثلثی باشد، مقدار ۱، وگرنه مقدار صفر را برمی‌گرداند. دوازده عدد مثلثی در زیر آمده‌اند:
- $$۰ \quad ۱ \quad ۳ \quad ۶ \quad ۱۰ \quad ۱۵ \quad ۲۱ \quad ۲۸ \quad ۳۶ \quad ۴۵ \quad ۵۵ \quad ۶۶$$
- متدی به نام isSquare که تشخیص می‌دهد، یک عدد مربعی است یا خیر؟ چند عدد مربعی عبارت‌اند از:
- $$۰ \quad ۱ \quad ۴ \quad ۹ \quad ۱۶ \quad ۲۵ \quad ۳۶ \quad ۴۹ \quad ۶۴ \quad ۸۱ \quad ۱۰۰ \quad ۱۲۱ \quad \dots$$
- برنامه‌ای که x و y را خوانده، حاصل عبارت  $\sqrt{|x-y|^{|y|}}$  را محاسبه کرده، نمایش می‌دهد.
- برنامه‌ای که نمره‌ای را به عدد بین صفر تا ۱۰۰ خوانده، توسط متدی مقدار حرفی معادل آن را برمی‌گرداند و نمایش می‌دهد (اگر نمره بین ۸۰ تا ۱۰۰ باشد، حرف A، اگر نمره بین ۶۰ تا ۸۰ باشد، حرف B، چنانچه نمره بین ۵۰ تا ۶۰ باشد، حروف C، وگرنه حروف F را نمایش می‌دهد).

## ۶۰۰ برنامه C# با حل آن‌ها ۶۷

➤ متد main: متغیر با نوع int به نام grade تعریف کرده، سپس با یک پیغام مناسب نمره را از ورودی خوانده، در متغیر grade قرار می‌دهد، در ادامه با پیام مناسب و فراخوانی متد selectGrade نمره حرفی را نمایش می‌دهد.

➤ متد selectGrade: نمره عددی را از طریق پارامتر grade دریافت می‌کند و نمره حرفی را برمی‌گرداند. چون نوع متد char انتخاب گردید، با استفاده از ifهای متداخل یکی از حروف A، B، C یا F را برمی‌گرداند (با توجه به مقدار پارامتر grade).

➤ برنامه‌ای که سن فردی را به سال، ماه و روز دریافت کرده، توسط متدهای به‌روز، ساعت، دقیقه و ثانیه تبدیل کرده، برمی‌گرداند و سپس نمایش می‌دهد (هرسال ۳۶۵٫۲۵ روز و هرماه ۳۰ روز است).

➤ برنامه‌ای که حقوق کارمند را خوانده، توسط متدی مالیات بر حقوق را محاسبه کرده، نمایش می‌دهد (مالیات از طریق جدول زیر محاسبه می‌گردد):

از	تا	درصد مالیات
0	483000	0
483001	600000	10%
600001	100000	15%
1000001	2000000	20%
2000001	99999999	30%

➤ برنامه‌ای که عددی را خوانده، با استفاده از متدی حاصل ضرب ارقام بالای ۵ آن را محاسبه کرده، نمایش می‌دهد.

➤ برنامه‌ای که عددی را خوانده، با استفاده از یک متد حاصل ضرب ارقام زوج آن‌ها را برمی‌گرداند و نمایش می‌دهد.

➤ برنامه‌ای که n را خوانده، با استفاده از یک متد n بار ۱۰ تا x را نمایش می‌دهد.

➤ برنامه‌ای که نمره پایان ترم و میان ترم دانشجویی را گرفته، با استفاده از یک متد نمره نهایی دانشجو را محاسبه و برمی‌گرداند (نمره نهایی دانشجو برابر با نمره میان ترم \* ۰٫۴ + نمره پایان ترم \* ۰٫۶ است).

➤ برنامه‌ای که استفاده از متغیرهای محلی و سراسری را نمایش می‌دهد.

➤ برنامه‌ای که اطلاعات جعبه‌ای از قبیل طول، عرض و ارتفاع را به‌عنوان پارامتر دریافت کرده، با استفاده از متدی حجم جعبه را محاسبه کرده، نمایش می‌دهد (هدف این برنامه، آشنایی بیش‌تر با متدها با پارامتری با مقادیر پیش‌فرض است). حجم جعبه برابر با ارتفاع  $\times$  عرض  $\times$  طول است.

➤ برنامه‌ای که قیمت کالا و درصد تخفیف را دریافت کرده، به متدی ارسال می‌کند و این متد میزان تخفیف کالا را برگردانده و در برنامه میزان تخفیف را نمایش می‌دهد.

➤ برنامه‌ای که میزان موجودی حساب بانکی را به همراه درصد بهره سالانه دریافت کرده، تعیین می‌کند سپس از چند سال موجودی حساب (بدون برداشت هیچ مبلغی از حساب) به مبلغ خاصی می‌رسد (موجودی انتهای هر سال توسط متدی محاسبه می‌گردد).

➤ برنامه‌ای که کاراکتری را از ورودی خوانده، توسط متدهای تعیین می‌کند که کاراکتر خوانده‌شده، رقم، حرف بزرگ و حرف کوچک است یا خیر؟

➤ برنامه‌ای که عدد  $n$  را خوانده، حاصل عبارت زیر را نمایش می‌دهد:

$$1! + 2! + \dots + n!$$

➤ این برنامه متدی برای محاسبه فاکتوریل و متد دیگری برای محاسبه مجموع دارد.

➤ برنامه‌ای که  $n$  را خوانده، حاصل عبارت مقابل را نمایش می‌دهد:

$$\frac{1}{1!} + \frac{2}{2!} + \frac{3}{3!} + \dots + \frac{n}{n!}$$

➤ برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل عبارت مقابل را نمایش می‌دهد:

$$\frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

➤ این برنامه متدهای برای محاسبه فاکتوریل، توان ( $x^n$ ) و مجموع سری دارد.

➤ برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل جمع  $n$  جمله سری زیر را نمایش می‌دهد:

$$\frac{x}{1} + \frac{x^2}{2} + \frac{x^3}{1*3} + \dots + \frac{x^5}{1*3*5}$$

➤ برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل  $n$  جمله سری زیر را نمایش می‌دهد:

$$\frac{x}{1} - \frac{x^2}{2} + \frac{x^3}{1*3} - \frac{x^4}{2*4} + \frac{x^5}{1*3*5} + \dots$$

➤ برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل  $n/2$  جمله عبارت زیر را نمایش می‌دهد:

## ۶۰۰ برنامه C# با حل آن‌ها ۶۹

$$-\frac{x^2}{2} + \frac{x^4}{2*4} - \frac{x^6}{2*4*6} + \dots$$

➤ برنامه‌ای که دو مقدار عددی را خوانده، سپس توسط متدی به عدد اول یک واحد اضافه نماید و از عدد دوم یک واحد کسر کرده، آن‌ها را در برنامه اصلی نمایش می‌دهد.

➤ برنامه‌ای که  $x$  و  $k$  را خوانده، لگاریتم  $x$  در مبنای  $k$  را با یک متد محاسبه کرده، نمایش می‌دهد.

➤ برنامه‌ای سه عدد را خوانده توسط متدی یکی از مقادیر زیر را برگرداند:  
اگر سه عدد تشکیل مثلث را نمی‌دهند، صفر را برگرداند.  
اگر سه عدد تشکیل مثلث قائم‌الزاویه را می‌دهند، عدد یک را برمی‌گرداند.  
اگر سه عدد تشکیل مثلث متساوی‌الساقین را می‌دهند، عدد ۲ را برمی‌گرداند.  
اگر سه عدد تشکیل مثلث متساوی‌الاضلاع را می‌دهند، عدد ۳ را برمی‌گرداند.  
وگرنه عدد ۱- را برمی‌گرداند.

➤ برنامه‌ای که اعداد  $n$  و  $b$  را خوانده، عدد  $n$  را به روش بازگشتی از مبنای ۱۰ به مبنای  $b$  تبدیل می‌کند (دقت داشته باشید که  $b$ ، مبنای کوچک‌تر از ۱۰ است).

➤ برنامه‌ای که با روش بازگشتی مقدار مشتق  $\frac{d(n)}{dn}(e^{ax})$  را محاسبه می‌کند:

توضیح: اگر از متد  $e^{ax}$  مشتق بگیریم، رابطه بازگشتی مقابل به دست می‌آید:

$$(e^{ax})^n = a(e^{ax})^{n-1}$$

➤ برنامه‌ای که  $\sin$ ،  $\cos$ ،  $\tan$  زوایای مضرب پنج از ۵ تا ۹۰ را در خروجی با فرمت جدول‌بندی مناسب تا سه رقم اعشار نمایش می‌دهد. برای محاسبه زوایا باید از درجه به رادیان تبدیل شوند. برای این منظور از فرمول زیر استفاده می‌شود:

$$\frac{R}{3.1415} = \frac{D}{180} \Rightarrow R = \frac{3.1415 * D}{180}$$

➤ برنامه‌ای که ۱۵ جمله دنباله زیر را به کمک بازگشتی زیر نمایش می‌دهد:

$$F(1) = 2 \quad \text{اگر } n == 1$$

$$F(n) = 2 * F(n-1) + 1 \quad \text{وگرنه}$$

➤ برنامه‌ای که حاصل عبارت  $f(x) = 5x^2 - 3x + 4$  را برای مقادیر صحیح و اعشاری  $x$  با دو متد هم نام محاسبه کرده و نمایش می‌دهد.

➤ برنامه‌ای که قاعده و ارتفاع مثلث را خوانده، با استفاده دو متد هم نام مساحت مثلث را حساب می‌کند. مساحت مثلث برابر با قاعده ضرب در نصف ارتفاع است. در ضمن ارتفاع و قاعده می‌توانند دو عدد صحیح یا دو عدد اعشاری باشند.

➤ برنامه‌ای که کاراکتری را خوانده، بدون استفاده از عملگر  $+$  و  $if$  آن را به حروف کوچک تبدیل می‌کند و نمایش می‌دهد. این برنامه برای تبدیل حروف بزرگ و کوچک از یک متد استفاده می‌کند.

➤ برنامه‌ای که یک کاراکتر را خوانده، با استفاده از عملگر  $xor$  (^) اگر حرف کوچک بود، آن را به حروف بزرگ تبدیل کرده، یا اگر حرف بزرگ بود، آن را به حرف کوچک تبدیل نماید. برای این منظور، از یک متد استفاده نماید.

➤ برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل متد  $F(x, n) = x^n$  را نمایش می‌دهد.

➤ برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل عبارت مقابل را نمایش می‌دهد

$$F(x, n) = \sum_{k=1}^n \frac{x^n}{k!} :$$

➤ برنامه‌ای که دو عدد صفر یا ۱ را خوانده، نتایج عملگرهای  $XOR$ ،  $OR$ ،  $AND$ ،  $NOR$ ،  $NAND$  را نمایش می‌دهد. در این برنامه متدهای برای برگشت نتایج عملگرهای  $XOR$ ،  $OR$ ،  $AND$ ،  $NOR$ ،  $NAND$  و پیاده‌سازی شوند.

➤ برنامه‌ای که  $n$  را خوانده، تمام اعداد تام (کامل) کوچک‌تر یا مساوی  $n$  که مضرب ۳ هم باشند را نمایش می‌دهد. (عددی تام است که مجموع مقسوم‌های کوچک‌تر از خودش برابر خودش باشد).

➤ متد بازگشتی که دو عدد  $x, y$  را خوانده، حاصل عبارت زیر را برمی‌گرداند:

$$F(x, y) = \begin{cases} x - y & \text{اگر } x \text{ یا } y \text{ کوچک‌تر از صفر باشد} \\ F(x-1, y) + F(x, y-1) & \text{وگرنه} \end{cases}$$

برنامه‌ای که از این متد بازگشتی استفاده می‌کند.

➤ متد بازگشتی  $n$  را به عنوان پارامتر دریافت کرده، تمام اعداد مضرب ۳ کوچک‌تر از  $n$  را نمایش می‌دهد. برنامه که از این متد استفاده می‌کند.

## ۶۰۰ برنامه C# با حل آن‌ها ۷۱

- متد بازگشتی که عددی را به عنوان پارامتر دریافت می‌کند و تمام اعداد فرد بین این دو عدد را نمایش می‌دهد. برنامه‌ای که از این متد استفاده می‌کند.
- موسسه‌ای قصد دارد  $x$  درصد به حقوق کارمندانش اضافه کند. متدی که ضریب افزایش حقوق و حقوق فعلی کارمند را دریافت کرده، حقوق جدید را برمی‌گرداند. برنامه‌ای که برای  $n$  کارمند از این متد استفاده می‌کند.
- متدی که دو عدد را به عنوان پارامتر دریافت کرده، تعیین می‌کند چند عدد اول بین این دو عدد وجود دارند. این برنامه، متدی برای تعیین عدد اول دارد. برنامه‌ای که از این متدها استفاده می‌کند. “ عدد اول است که بر هیچ عدد کوچک‌تر از نصف خودش به جزء یک بخش پذیر نباشد.”
- متدی که تأثیر نیروی جاذبه بر اجسام در حال سقوط را محاسبه می‌کند. برنامه‌ای که زمان را دریافت کرده (برحسب ثانیه) و این تأثیر را محاسبه می‌نماید و چاپ می‌کند. این برنامه باید ارتفاع شیء سقوط کننده در هر ثانیه را نمایش می‌دهد. برای این منظور از فرمول‌های زیر استفاده می‌نماید:  
$$\text{زمان } t = 9,80665, g = \text{فاصله و ارتفاع } t^2 * g * \frac{1}{2} = \text{فاصله}$$
- برنامه‌ای که عدد  $n$  را خوانده، سپس یک عدد دیگر  $k$  را بخواند و حاصل ضرب اعداد کوچک‌تر یا مساوی عدد  $n$  که مضرب آن عدد  $k$  باشد را نمایش می‌دهد.
- اگر دو ضلع  $(B, C)$  از مثلث و زاویه بین این دو ضلع  $\text{Alpha}$  باشد، می‌توانیم طول ضلع سوم را با فرمول زیر محاسبه کنیم:  
$$A^2 = B^2 + C^2 - 2 * B * C * \cos(\text{Alpha})$$
- برنامه‌ای که  $B, C$  و  $\text{Alpha}$  را خوانده، با استفاده از متدی مقدار  $A$  را محاسبه می‌کند و نمایش می‌دهد.
- برنامه‌ای که دو عدد، سپس یک کاراکتر را خوانده، اگر کاراکتر  $R$  یا  $r$  باشد، مساحت مستطیل را با یک متد محاسبه می‌کند. چنانچه کاراکتر  $T$  یا  $t$  را وارد نماید، مساحت مثلث را با فرمول ارتفاع \* نصف قاعده محاسبه می‌نماید. در نهایت مساحت مستطیل یا مثلث را نمایش می‌دهد.
- فرض کنید اساتید دارای حقوق کم‌تر از ۴۰۰۰۰۰۰ تومان ۰/۵ درصد اضافه حقوق می‌گیرند، ولی اساتید بین ۴۰۰۰۰۰۰ تا ۷۰۰۰۰۰۰ تومان ۱۲ درصد و اساتید با حقوق بیش‌تر ۱۰ درصد اضافه حقوق می‌گیرند. متدی که اضافه حقوق استاد را محاسبه می‌کند. برنامه‌ای که حقوق استادی را خوانده، اضافه حقوق و حقوق جدید استاد را نمایش می‌دهد.

➤ برنامه‌ای که  $n$  و  $m$  را خوانده، توسط متد بازگشتی  $C_n^m$  را برگرداند، نمایش می‌دهد:

$$C_n^m = \begin{cases} 1 & \text{اگر } m = 0 \text{ یا } m = n \\ n & \text{اگر } m = 0 \\ \binom{n-1}{m} + \binom{n-1}{m-1} & \text{وگرنه} \end{cases}$$

➤ برنامه‌ای که عددی را خوانده با استفاده از متد بازگشتی مغلوب آن را نمایش می‌دهد. به‌عنوان مثال، اگر عدد ۳۲۵۷ باشد، مغلوب آن یعنی ۷۵۲۳ را نمایش می‌دهد.

➤ برنامه‌ای که  $n$  و  $m$  را خوانده، توسط یک متد بازگشتی  $A(m, n)$  را محاسبه می‌کند و نمایش می‌دهد  $A(m, n)$  به‌صورت زیر محاسبه می‌شود.

$$A(m, n) = \begin{cases} n + 1 & \text{اگر } m = 0 \\ A(m-1, 1) & \text{اگر } n = 0 \\ A(m-1, A(m, n-1)) & \text{وگرنه} \end{cases}$$

➤ برنامه‌ای که  $n$  و  $m$  را خوانده، توسط یک متد بازگشتی  $Q(m, n)$  را محاسبه می‌کند و

$$Q(m, n) = \begin{cases} 1 & \text{وگرنه } n = 1 \text{ یا } m = 1 \text{ باشد} \\ Q(m, n) & \text{اگر } m < n \text{ باشد} \\ 1 + Q(m, m-1) & \text{اگر } m = n \text{ باشد} \\ Q(m, n-1) + Q(m-n, n) & \text{اگر } m > n \text{ باشد} \end{cases}$$

نمایش می‌دهد:

➤ برنامه‌ای که مشتق متد  $1 + x^2$  را در نقطه  $x$  محاسبه می‌کند. در این برنامه به ازای  $h$  کوچک می‌توان نوشت:

$$f'(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

➤ برنامه‌ای که مشتق دوم متد  $1 + x \sin x$  در نقطه  $x$  با فرمول زیر محاسبه می‌کند.

$$f''(x_0) = \lim_{h \rightarrow 0} \frac{f'(x_0 + h) - f'(x_0)}{h} = \frac{f'(x_0 + h) - f'(x_0)}{h}$$

➤ برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\left(1 + \frac{x^n}{1}\right)^1 + \left(1 + \frac{x^{(n-1)}}{2}\right)^2 + \dots + \left(1 + \frac{x^1}{n}\right)^n$$

➤ برنامه‌ای که سه عدد  $x$ ،  $n$  و  $a$  را دریافت کرده، حاصل عبارت زیر را محاسبه می‌کند و نمایش می‌دهد:

$$(x+a)^n = \frac{n!}{0!(n-0)!} x^0 a^{n-0} + \frac{n!}{1!(n-1)!} x^1 a^{n-1} + \dots + \frac{n!}{n!(n-n)!} x^n a^{n-n}$$

➤ برنامه‌ای که اعداد صحیح x و n را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\begin{aligned} & n! \frac{x^n}{1} + (n-1)! \frac{x^{(n-1)}}{1 - \frac{1}{x}} + (n+1)! \frac{x^{(n+1)}}{1 + \frac{1}{x}} \\ & + (n-2)! \frac{x^{(n-2)}}{2 - \frac{2}{x}} + (n+2)! \frac{x^{(n+2)}}{2 + \frac{2}{x}} + \dots + \\ & x^{(n-n)} \frac{(n-n)!}{n - \frac{n}{x}} + x^{(n+n)} \frac{(n+n)!}{n + \frac{n}{x}} \end{aligned}$$

➤ پروژه برنامه‌نویسی ۱. برنامه‌ای که تمامی جفت‌های اعداد متحابه از یک تا عدد خوانده‌شده را نمایش می‌دهد. دو عدد صحیح متحابه گویند، اگر مجموع مقسوم‌علیه‌های هر یک به‌جز خود عدد با عدد دوم برابر باشد. به‌عنوان مثال، مقسوم‌علیه‌های ۲۲۰ که از ۲۲۰ کوچک‌تر هستند، عبارت انداز: ۵۵، ۱۱۰، ۲۲، ۱۰، ۵، ۴، ۲، ۱ که مجموع آن‌ها عدد ۲۴۸ می‌شود و مقسوم‌علیه‌های ۲۴۸ که کوچک‌تر از ۲۴۸ هستند، برابر با ۱۴۲، ۷۱، ۴، ۲، ۱ که مجموع آن‌ها ۲۲۰ است. پس این دو عدد متحابه می‌باشند.

➤ پروژه برنامه‌نویسی ۲. برنامه‌ای که مقداری را از یک مبنا دریافت کرده، به مبنا دیگری تبدیل می‌کند (مبناها می‌توانند بین ۲ تا ۱۶ باشند). این پروژه مبنا می‌باشد، عددی در آن مبنا (چنانچه کاربر مقدار را خارج از مبنا وارد کند، جلوی او را می‌گیرد) و مبنا مقصد را از کاربر دریافت می‌کند.

➤ پروژه برنامه‌نویسی ۳. برنامه‌ای که ابتدا منوی زیر را نشان می‌دهد:

Calculate one  
Calculate two  
Calculate three  
Exit  
SELECT (۱ to ۴):

اگر کاربر عدد ۱ را وارد کند، حاصل عبارت زیر را محاسبه می‌کند و نمایش می‌دهد:

$$\frac{x^n (n)^5 (n)^{x+2}}{x! n! (x+n)!} \sum_{i=1}^n \frac{x^i i^x}{x! n! (n+2)!}$$

اگر کاربر عدد ۲ را وارد کند، حاصل عبارت زیر را محاسبه می‌کند و نمایش می‌دهد:

$$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \pm \frac{x^n}{n!}$$

اگر کاربر عدد ۳ را وارد کند، حاصل عبارت زیر را محاسبه می‌کند و نمایش می‌دهد:

$$\frac{((x)^n)^3 (n)^{x+3}}{n! + (x+n+2)!} \sum_{i=2}^x \frac{x^i i^x n^x}{(x+i)! + (n+x)!} \quad i = 2, 4, \dots$$

اگر کاربر عدد ۴ را وارد کند، برنامه خاتمه یابد.

➤ پروژه برنامه نویسی ۴. فرض کنید که سیستم را برای یک بنگاه معاملات ملکی می نویسد. در این سیستم قرار است، منوی با ۵ گزینه نمایش داده شود تا کاربر با انتخاب یکی از گزینه ها شکل زمین فروخته شده را انتخاب کند. زمین ها به شکل های مربع، مستطیل، مثلث و دایره هستند. پس از انتخاب شکل زمین توسط کاربر، قیمت هر متر زمین و درصد حق السهم فروشنده دریافت می گردد و در ادامه با توجه به نوع انتخاب شده کاربر، اطلاعات زمین دریافت می گردد و مبلغی که خریدار باید پرداخت کند را نمایش می دهد. این برنامه تا زمانی ادامه می یابد که کاربر در منوی ظاهر شده گزینه ۵ را انتخاب نماید.

➤ برنامه‌ای که  $n$  عدد را خوانده، در آرایه‌ای قرار می‌دهد و توسط متدی عناصر آن را معکوس (از آخرین عدد به اولین عدد) کرده، به برنامه برگرداند و برنامه آن را نمایش می‌دهد (برنامه، برای دریافت داده، معکوس کردن و چاپ عناصر آرایه از متدها جداگانه استفاده می‌کند).

➤ فرض کنید که در قسمت تخلفات ماشین‌ها در اداره راهنمایی رانندگی، ۱۰ نوع تخلف منظور گردید و کد تخلف از ۰ تا ۹ در نظر گرفته شد و هر تخلف جریمه خاصی دارد. برای هر ماشین اطلاعاتی مثل شماره ماشین، تعداد تخلفات و کد هر تخلف موجود است. برنامه‌ای که مبلغ جریمه را برای هر ماشین محاسبه می‌کند. برای خروج از برنامه کاربر به جای شماره ماشین ۹۹۹- را وارد می‌کند.

توضیح: برای حل این برنامه کد تخلف به همراه مبلغ جریمه را در آرایه‌ای به نام code قراردادیم که ۱۰ سطر دارد. در هر سطر مبلغ جریمه و کد تخلف نگهداری می‌شود.

➤ برنامه‌ای که عناصر آرایه‌ای  $4 \times 4$  را خوانده، بزرگ‌ترین عنصر هر سطر را پیدا کرده، به همراه شماره سطر در خروجی چاپ می‌کند.

توضیح: ستون پنجم آرایه برای نگهداری بزرگ‌ترین عنصر هر سطر در نظر گرفته شده است.

➤ برنامه‌ای که رشته‌ای را در رشته دیگر کپی می‌کند (بدون استفاده از متد strcpy).

➤ ماتریس جادویی یک ماتریس  $N \times N$  است که هر عنصر آن یک عدد صحیح ۱ تا  $N^2$  است. حاصل جمع هر سطر، هر ستون و همچنین عناصر قطر همگی با یک دیگر مساوی‌اند. برای تولید این ماتریس به صورت زیر عمل می‌شود:

الف. عدد یک در ستون میانی اولین سطر قرار می‌گیرد.

ب. به طور مورب به سمت چپ و بالا حرکت می‌کنیم.

ج. اگر این خانه پر باشد، یک خانه به طرف پائین حرکت می‌کنیم و عدد بعدی را در آن مکان می‌نویسیم.

د. اگر در حرکت به صورت مورب، از محدوده ماتریس خارج شویم، باید به آخرین عنصر در آن سطر یا ستون برگردیم.

هـ اگر در محدوده سطر و ستون از هر دو خارج شویم، یک خانه به سمت پائین حرکت می‌کنیم.  
 برنامه‌ای که ماتریس جادویی را تولید کرده و چاپ می‌نماید (ماتریس  $5 \times 5$  زیر را در نظر بگیرد):


- برنامه‌ای که رشته‌ای را به انتهای رشته دیگر الحاق می‌کند (بدون استفاده از متد Strcat).
- برنامه‌ای که رشته‌ای را از ورودی خوانده، به متدی ارسال می‌کند و متد آن را به‌طور معکوس به خروجی می‌برد.
- برنامه‌ای که رشته عددی را که حاوی نقطه اعشار است از ورودی خوانده، آن را به عدد اعشاری تبدیل کند. به‌عنوان مثال، رشته "۱۲,۳۴۲" را به ۱۲,۳۴۲ تبدیل نماید. این برنامه، متدی برای خواندن رشته، متدی برای تبدیل و متدی برای نوشتن عدد در خروجی دارد. پارامترها از طریق فراخوانی با ارجاع به متدها ارسال می‌شوند.
- برنامه‌ای که بازی دوز را شبیه‌سازی می‌کند. در این بازی یک ماتریس  $3 \times 3$  داریم که دو بازیکن ۱ و ۲ باهم بازی می‌کنند که هر یک از این بازیکنان سه مهره دارند (هر سه مهره شماره همان بازیکن را دارد). بازیکنی برنده است که بتواند سریع‌تر مهره‌های خود را به‌صورت سطری یا ستونی و یا قطری ردیف کند. به‌عنوان مثال، در ماتریس زیر بازیکن شماره ۲ برنده است:
- هر عدد زوج بزرگ‌تر از ۶ برابر با مجموع دو عدد اول کوچک‌تر از خودش است. برنامه‌ای که چند عدد زوج بزرگ‌تر از ۶ را می‌خواند. مجموع دو عدد اولی که برابر با اعداد خوانده شده باشد را پیدا کرده چاپ می‌نماید (برنامه تا زمانی که کاربر بخواهد ادامه می‌یابد).

## ۶۰۰ برنامه C# با حل آن‌ها ۷۷

➤ برنامه‌ای که رشته عددی را که حاوی نقطه اعشار است از ورودی خوانده، آن را به صورتی که گفته شد به عدد اعشاری تبدیل می‌کند. به عنوان مثال، رشته '۱۲۴۵,۴۵۲' را به عدد ۱۲۴۵,۴۵۲ تبدیل می‌کند. برنامه، متدی برای خواندن رشته، متدی برای تبدیل و متدی برای نوشتن عدد در خروجی دارد. پارامترها از طریق فراخوانی با ارجاع به متدها ارسال می‌شوند.

➤ متدی که طول رشته را برمی‌گرداند (با استفاده از اشاره گر). سپس، برنامه‌ای که از این متد استفاده می‌کند. برنامه، متدی برای خواندن رشته، متدی برای محاسبه طول رشته و متدی برای چاپ طول رشته دارد.

➤ متدی که توان  $n$  یک عدد را برمی‌گرداند. سپس، برنامه‌ای که از این متد استفاده می‌کند.

➤ برنامه‌ای که سه رشته را خوانده، رشته دوم و سوم را به انتهای رشته اول متصل می‌کند.

➤ برنامه‌ای که رشته‌ای را خوانده، اعمال زیر را انجام می‌دهد:  
الف. کلیه حروف بزرگ رشته را به حروف کوچک تبدیل می‌کند.  
ب. کلیه ارقام رشته را شمارش می‌کند و مجموع ارقام موجود در رشته را چاپ می‌کند.  
ج: تعداد حروف رشته را شمارش می‌کند.  
برنامه، برای هر یک از این درخواست‌ها متدی دارد و کلیه پارامترهای متد را با ارجاع ارسال می‌نماید.

➤ برنامه‌ای که میزان میانگین بارش باران را در سال قبل می‌گیرد و سپس میزان واقعی بارش باران را در ۱۲ ماه گرفته، در آرایه‌ای قرار می‌دهد (برای خواندن میزان واقعی بارش باران در ۱۲ ماه از یک متد استفاده می‌کند). سپس، با یک متد دیگر اختلاف بارش باران در هرماه را با میانگین بارش باران در سال قبل محاسبه کرده، نمایش می‌دهد.

➤ برنامه‌ای که ۵ عدد را از ورودی خوانده، در آرایه‌ای به نام  $S$  قرار می‌دهد. ابتدا میانگین مقادیر را محاسبه می‌کند (با یک متد). سپس، با استفاده از فرمول زیر انحراف معیار را محاسبه کرده، چاپ می‌نماید (برای محاسبه انحراف معیار نیز از یک متد استفاده

$$\text{انحراف معیار} = \frac{\sqrt{\sum_{i=1}^N (S_i - a)^2}}{N}$$

می‌کند):

$a$  میانگین اعداد و  $N$  تعداد اعداد است.

➤ برنامه‌ای که رشته‌ای را از ورودی می‌خواند. تمام کلمات چهارحرفی آن را با کلمه "Love" جایگزین می‌کند. مثلاً رشته "hate you, you doer" به رشته "love you, you love" تبدیل می‌گردد.

➤ در یک دوره مسابقات ورزشی تعدادی تیم (حداکثر ۱۰۰ تیم) شرکت کرده‌اند، می‌خواهیم در انتهای مسابقه گزارشی از مجموع امتیازات هر تیم را چاپ کنیم. برنامه‌ای که برای هر مسابقه در یک خط ورودی شامل آیت‌های برنده و امتیاز کسب‌شده را می‌خواند و نتیجه را نمایش می‌دهد. اگر به جای تیم برنده ۱- وارد شد، برنامه خاتمه می‌یابد.

➤ برنامه‌ای که یک رشته را خوانده، تعداد تکرار هر حرف را نمایش می‌دهد. به عنوان مثال، ACCURENCE را در نظر بگیرید که حرف C سه بار تکرار شده است.

➤ برنامه‌ای که یک رشته را خوانده تشخیص می‌دهد آیا متجانس است یا خیر؟ رشته‌ای متجانس است که از دو طرف یکی خوانده شود.

➤ برنامه‌ای که حداکثر ۱۰ عدد را از ورودی خوانده، می‌گوید هر عدد چند بار تکرار شده است. به عنوان مثال، اگر ورودی به صورت زیر باشد:

۷۰ ۱۴۳ ۱۰۰ ۷۲ ۱۴۳ ۵۲ ۱۰۰ ۱۴۳ ۱۰۰ ۷۰

خروجی زیر را چاپ می‌کند:

۱۴۳ OCCURS ۳ TIMES, AT POSITIONS ۳ ۶ ۹  
۷۰ OCCURS ۲ TIMES, AT POSITIONS ۱ ۱۰

➤ برنامه‌ای ماتریس  $A[3 \times 4]$  را از ورودی می‌خواند و ترانواده آن را در B ذخیره می‌کند. در ترانواده باید برای هر  $i, j$  رابطه مقابل برقرار باشد:

$$B_{i,j} = A_{j,i}$$

➤ برنامه‌ای که انداختن دو تاس را شبیه‌سازی می‌نماید. این برنامه برای انداختن تاس‌ها از متد rand استفاده می‌کند که عددی بین ۱ تا ۶ را تولید می‌نماید. سپس مجموع این دو مقدار محاسبه می‌گردد. از آنجایی که تاس‌ها مقادیر بین ۱ تا ۶ را نشان می‌دهند، پس مجموع این دو مقدار بین ۲ تا ۱۲ است که مجموع ۷، بیش‌ترین دفعات و مجموع ۱۲ کم‌ترین دفعات پیش می‌آیند. جدول زیر ترکیبات ممکن برای این دو تاس را نشان می‌دهد. این برنامه، ۳۶۰۰۰ بار تاس می‌ریزد و با استفاده از یک آرایه، تک‌بعدی تعداد دفعاتی را که هر یک از این مجموعه‌های ممکن پیش می‌آیند، را ثبت می‌کند و سپس تحقیق می‌کند که آیا تعداد مجموعه‌های حاصل معقول است (مثلاً به شش طریق، مجموع ۷ حاصل می‌شود. بنابراین تقریباً  $1/6$  همه پیشامدها باید ۷ باشد).

۱ ۲ ۳ ۴ ۵ ۶

## ۶۰۰ برنامه C# با حل آن‌ها ۷۹

۱	۲	۳	۴	۵	۶	۷
۲	۳	۴	۵	۶	۷	۸
۳	۴	۵	۶	۷	۸	۹
۴	۵	۶	۷	۸	۹	۱۰
۵	۶	۷	۸	۹	۱۰	۱۱
۶	۷	۸	۹	۱۰	۱۱	۱۲

➤ در هر سطر ورودی برای هر کالا دو عدد وجود دارد. عدد اول شماره کالا و عدد دوم موجودی آن در انبار است.

الف. برنامه‌ای که این اطلاعات را می‌خواند و در یک آرایه ذخیره می‌کند. انتهای اطلاعات با دو عدد ۱- مشخص شده است.

ب: پس از ورود اطلاعات اولیه تعدادی ورودی دیگر وجود دارد. این ورودی‌ها عبارت‌اند از: شماره کالا، تعداد و کاراکتری که مشخص‌کننده این است که تعداد ورودی سفارش (کاراکتر s یا S) یا خرید (کاراکتر p یا P) است. برنامه‌ای که اطلاعات را خوانده، مقدار خروجی انبار هر کالا را نمایش می‌دهد. اگر سفارش بیش از موجودی انبار باشد، آن را انجام نمی‌دهد و پیام مناسبی چاپ می‌کند.

➤ برنامه‌ای که تعدادی کلمه را از ورودی خوانده، تعداد کلمات ۱ تا ۵ کاراکتری، ۶ تا ۱۲ کاراکتری و بیش‌تر از ۱۲ کاراکتر را نمایش می‌دهد (تعداد کلمات را از ورودی می‌خواند).

➤ برنامه‌ای که چندجمله‌ای زیر را در یک آرایه نمایش می‌دهد (سپس، x را می‌خواند) و عبارت p را محاسبه کرده، چاپ می‌نماید (n حداکثر ۱۰۰ باشد).

$$p = anx^n + an - 1x^{n-1} + \dots + a_1x^1 + a.$$

➤ برنامه‌ای که یک رشته حداکثر ۱۰۰ کلمه‌ای را خوانده، تمام کلمات آن را وارونه می‌کند و چاپ می‌نماید (بین کلمات کاراکتر blank (فاصله) قرار می‌گیرد).  
به عنوان مثال، Ali and Reza به صورت R e z a   d n a   i l A تبدیل می‌شود.

➤ برنامه‌ای که یک متن را از ورودی خوانده، اعمال زیر را انجام می‌دهد:  
الف: درصد تکرار هر حرف را در متن پیدا می‌کند (حروف بین A تا Z و حروف کوچک را بزرگ در نظر بگیرید).

ب: درصد تکرار هر کلمه را در متن پیدا می کند (کلمات بافاصله از یکدیگر جدا می شوند (حداکثر ۱۰۰ کلمه داریم).

➤ برنامه ای که تعدادی عدد را خوانده، تشخیص می دهد مرتب شده هستند یا خیر؟

➤ برنامه ای که  $n$  عدد را خوانده، در آرایه قرار می دهد. سپس، توسط متدی عنصر  $n$  را حذف می کند.

➤ برنامه ای که دو آرایه  $m$  عنصری و  $n$  عنصری را خوانده،  $n$  عنصر اول آرایه دوم را به انتهای آرایه اول اضافه می کند.

➤ برنامه ای که عناصر یک آرایه و مقدار  $x$  را خوانده، تعداد تکرار عنصر  $x$  در آرایه  $a$  را می شمارد. در این برنامه متدها زیر نوشته شده است:  
متدی برای خواندن عناصر آرایه  $a$   
متدی برای شمارش تعداد تکرار  $x$  در آرایه  $a$

➤ برنامه ای که دو آرایه  $n$  عنصری یک بعدی را خوانده، حاصل ضرب بیرونی  $n$  عنصر اول آرایه  $a$  با  $n$  عنصر اول آرایه  $b$  را برمی گرداند (حاصل ضرب این دو آرایه، در آرایه دیگری به نام  $c$  قرار می گیرد). به عنوان مثال، فرض کنید آرایه های زیر را داشته باشیم.

a	b			
2.2	2.0	4.4	-2.2	0.0
3.3	-1.0	6.6	-3.3	0.0
4.4	0.0	8.8	-4.4	0.0

متدی برای ضرب بیرونی (outer product) داشته باشید.

متدی برای چاپ حاصل ضرب داشته باشید.  
 $C[i][j] = a[i] \times b[j]$

➤ برنامه ای که عناصر یک آرایه دوبعدی را ۹۰ درجه چرخش می دهد. به عنوان مثال، اگر آرایه زیر را داشته باشیم:

۱۱	۲۲	۳۳
۴۴	۵۵	۶۶
۷۷	۸۸	۹۹

با چرخش ۹۰ درجه آرایه به صورت زیر تبدیل می شود:

۷۷	۴۴	۱۱
----	----	----

## ۶۰۰ برنامه C# با حل آن‌ها ۸۱

۸۸	۵۵	۲۲
۹۹	۶۶	۳۳

➤ برنامه‌ای که آرایه‌ای ۱۰ عنصری تعریف کرده، سپس ۱۰ عدد را خوانده و تعداد تکرار هر عدد را می‌شمارد.

➤ برنامه‌ای که ۵ عدد را خوانده، در آرایه‌ای قرار می‌دهد. سپس یک عدد دیگر خوانده، تعداد تکرار این عدد و مکان‌های وقوع آن را نمایش می‌دهد.

➤ برنامه‌ای که ۱۰ عدد را خوانده، در آرایه‌ای قرار می‌دهد. سپس یک عدد دیگر را می‌خواند و چنانچه آن عدد در آرایه وجود داشته باشد، آن عدد را از آرایه حذف کرده، آرایه را نمایش می‌دهد (دقت کنید که اگر عدد چند بار در آرایه تکرار شده باشد، همه تکرارهای آن عدد را از آرایه حذف می‌کند).

➤ برنامه‌ای که یک آرایه  $4 \times 4$  تعریف کرده، سپس عناصر آرایه را خوانده و حاصل ضرب عناصر غیر صفر محیطی آرایه را محاسبه می‌کند و نمایش می‌دهد. عناصر محیطی آرایه، عناصری هستند که در سطر اول، سطر آخر، ستون اول و ستون آخر آرایه می‌باشند.

➤ برنامه‌ای که یک آرایه ۵ در ۶ تعریف کرده، عناصر آن را از ورودی می‌خواند و سپس حاصل ضرب عناصر غیر محیطی که صفر نیستند را محاسبه می‌کند و نمایش می‌دهد. عناصر غیر محیطی عناصری هستند که در سطر اول، سطر آخر، ستون اول و ستون آخر نباشند.

➤ برنامه‌ای که یک ماتریس  $4 \times 4$  را می‌خواند. اگر مجموع عناصر روی قطر اصلی و فرعی برابر مجموع عناصر محیطی ماتریس باشد، برنامه "Yes"، وگرنه "No" را نمایش می‌دهد. عناصری روی قطر اصلی قرار دارند، که  $i = j$  باشد (یعنی، شماره سطر برابر شماره ستون باشد) و عناصری روی قطر فرعی قرار دارند که  $i + j = n - 1$  باشد (یعنی، مجموع شماره سطر و شماره ستون برابر با تعداد سطرها منتهی یک باشد. اما، عناصر سطر اول، سطر آخر، ستون اول و ستون آخر عناصر محیطی نام دارند).

➤ برنامه‌ای که یک ماتریس  $4 \times 4$  را خوانده، اگر حداقل یک عنصر ماتریس تکراری باشد (بیش از یک‌بار تکرار شده باشد)، عبارت "Yes" وگرنه، عبارت "No" را نمایش می‌دهد.

➤ برنامه‌ای که یک ماتریس  $4 \times 4$  را خوانده، اگر همه عناصر آرایه برابر باشند، "Yes"، وگرنه "No" را نمایش می‌دهد.

➤ برنامه‌ای که یک ماتریس  $4 \times 4$  را خوانده، اگر تمام عناصر غیر از عناصر محیطی آن باهم برابر باشند، "Yes"، وگرنه "No" را نمایش می‌دهد.

- برنامه‌ای که عناصر آرایه  $4 \times 4$  را خوانده، اگر عناصر هر سطر به صورت صعودی مرتب باشند، "Yes"، وگرنه "No" را برمی‌گرداند.
- برنامه‌ای که یک آرایه  $4 \times 4$  را می‌خواند. اگر تمام عناصر آن به صورت صعودی مرتب باشند، عبارت "Yes"، وگرنه "No" را نمایش می‌دهد.
- برنامه‌ای که یک ماتریس  $4 \times 4$  را خوانده، مجموع عناصر قطر اصلی و فرعی را نمایش می‌دهد. (در عناصر قطر اصلی آرایه  $z == i$  است و اگر عنصری روی قطر فرعی باشد،  $i - 1 = n - z$  است.  $i$  شماره سطر و  $z$  شماره ستون آرایه دوبعدی است).
- برنامه‌ای که عناصر یک ماتریس  $5 \times 5$  را خوانده، بزرگ‌ترین عنصر محیطی آن را برمی‌گرداند.
- برنامه‌ای که یک رشته و عدد  $k$  را خوانده، تمام حروف الفبای رشته را به صورت زیر رمز کرده، نمایش می‌دهد.  

$$26\% (k + \text{حرف اصلی}) = \text{حرف رمز شده}$$

$$\%, \text{باقی مانده تقسیم صحیح به } 26 \text{ است.}$$
- برنامه‌ای که متن رمز شده و عدد  $k$  را خوانده، متن ساده را با روش زیر به دست می‌آورد و نمایش می‌دهد (یعنی، تمام حروف الفبایی را به صورت زیر رمزگشایی می‌کند):  

$$26\% (k - \text{حرف رمز شده}) = \text{حرف ساده}$$
- برنامه‌ای که تعدادی عدد را خوانده، در آرایه‌ای قرار می‌دهد. سپس مجموع اعداد اول در این اعداد را محاسبه می‌کند و در پایان، اعدادی که بزرگ‌تر از مجموع اعداد اول هستند را چاپ می‌کند.
- برنامه‌ای که ۱۰ عدد تصادفی بین ۱ تا ۴۰ تولید کرده، در آرایه اول قرار می‌دهد. سپس، ۱۰ عدد تصادفی دیگر بین ۱ تا ۵۰ تولید نموده، در آرایه دوم قرار می‌دهد. در پایان، عناصری که در آرایه اول وجود دارند، ولی در آرایه دوم نیستند را نمایش می‌دهد. یعنی تفاضل دو آرایه را نمایش می‌دهد.
- برنامه‌ای که ۱۰ عدد را خوانده، در آرایه‌ای قرار می‌دهد. سپس یک عدد بین ۰ تا ۸ گرفته، عناصر آرایه را به تعداد عدد خوانده شده به سمت چپ چرخش می‌دهد و چاپ می‌کند.
- برنامه‌ای که یک رشته را خوانده، تمام کلمات دوحرفی رشته را حذف کرده، آن را نمایش می‌دهد.

- برنامه‌ای که یک رشته را خوانده، تعداد کلمات یک حرفی، دو حرفی، سه حرفی، چهار حرفی و غیره را می‌شمارد.
- برنامه‌ای که ۱۰ عدد را خوانده، در آرایه‌ای قرار دهد. سپس تعیین می‌کند چه عددی بیش از همه تکرار شده است.
- برنامه‌ای که یک رشته را خوانده، کلیه جاهای خالی از قبیل کاراکتر blank، '\n' و '\t' رشته را حذف می‌کند و نمایش می‌دهد.

- یک عدد صحیح اول است، وقتی که بر خودش و یک بخش پذیر باشد. غربال اراتوستن روشی را برای تعیین عدد اول دارد. در این روش آرایه‌ای در نظر می‌گیرد و اعمال زیر را بر روی آن انجام می‌دهد تا اعداد اول را تعیین کند:
  ۱. مقدار اولیه همه عناصر آرایه را برابر یک قرار می‌دهد.
  ۲. با شروع از اندیس ۲ (اندیس یک حتماً اول است). هر بار که عنصری با مقدار یک یافت می‌شود، بقیه عناصری که اندیشان مضربی از ۲ باشد (اندیس‌های ۴، ۶، ۸، ۱۰ و ...) را برابر صفر قرار می‌دهد. برای اندیس ۳، همه عناصر بعد از ۳ که اندیشان مضربی از ۳ است (اندیس‌های ۶، ۹، ۱۲، ۱۵ و ...) صفر قرار می‌گیرد و به همین ترتیب این روند ادامه می‌یابد.
  ۳. در پایان عناصری که هنوز یک هستند، شماره اندیس‌هایشان اول اند.
- برنامه‌ای که با استفاده از یک آرایه ۱۰۰۰ تایی، اعداد اول ۱ تا ۹۹۹ را به دست آورده، نمایش می‌دهد.

- برنامه‌ای که آرایه‌ای ۱۰ عنصری با مقادیر اولیه را تعریف کرده و به روش بازگشتی مجموع عناصر آرایه را محاسبه و نمایش می‌دهد.
- برنامه‌ای که مقادیر آرایه ۱۰ عنصری که مقدار اولیه دارد، m واحد به سمت راست شیف می‌دهد.

- متد بازگشتی که ترکیبات یک مجموعه را برمی‌گرداند و نمایش می‌دهد. این متد دارای سه پارامتر است که عبارت‌اند از: شروع آرایه (مجموعه)، تعداد عناصر مجموعه و ترکیبات مجموعه. سپس، برنامه‌ای که تعداد ترکیبات مجموعه را نمایش می‌دهد. به عنوان مثال، مجموعه {A, B, C} را در نظر بگیرید. برای این مجموعه حالت‌های زیر را داریم:
 

{AAA, AAB, AAC, ABA, ABB, ABC, ACB, ACC, ...}
- همان‌طور که می‌دانید فاکتوریل اعداد بزرگ در یک double جا نمی‌شود. برنامه‌ای که فاکتوریل اعداد بزرگ را به کمک آرایه محاسبه می‌کند و نمایش می‌دهد.

➤ برنامه‌ای که دو آرایه  $3 \times 4$  به نام‌های  $x$  و  $y$  را خوانده، حاصل عبارت زیر را در آرایه دیگر به نام  $z$  قرار می‌دهد:

$$Z = 2x - 3y$$

➤ برنامه‌ای که بین تعدادی عدد که در آرایه قرار دارند، بزرگ‌ترین، کوچک‌ترین مقدار و مکان‌های آن‌ها را پیدا کرده، نمایش می‌دهد.

➤ برنامه‌ای که ۲۰ عدد تصادفی (دو سری اعداد تصادفی ۱۰ تایی) ایجاد کرده، در دو آرایه قرار می‌دهد. سپس یک عدد را خوانده، تمام عناصر آرایه دوم را در آرایه اول از مکان واردشده درج می‌نماید.

➤ برنامه‌ای که یک رشته را خوانده، سپس دو عدد را می‌خواند و از کاراکترهای عدد اول واردشده تا عدد دوم واردشده رشته ورودی را حذف می‌کند (یعنی، بخشی از رشته را حذف می‌نماید).

➤ برنامه‌ای که تعداد اعداد تصادفی تولید کرده، در آرایه  $4 \times 4$  قرار می‌دهد. سپس هر سطر آرایه را جداگانه مرتب می‌نماید. در پایان، آرایه‌ای که عناصر سطرهای آن مرتب‌شده‌اند را نمایش می‌دهد.

➤ برنامه‌ای که آرایه ۵ عنصری را خوانده، به مقدار هر یک از عناصر آرایه توسط علامت ستار (\*) میله افقی رسم می‌کند.

➤ برنامه‌ای که ۷ عدد را خوانده، سپس دو عدد دیگر را می‌خواند و از این دو عدد خوانده‌شده، عدد دوم را جایگزین عدد اول در آن هفت عدد می‌نماید.

➤ برنامه‌ای که عدد صحیح  $n$  (کوچک‌تر از ۹) را از ورودی خوانده، خروجی زیر را با استفاده از آرایه دوبعدی (ماتریس) تولید و چاپ می‌کند:

1	2	3	...	$n$
2	3	4	...	$n+1$
...	...	...	...	...
$n+1$	$n+2$	$n+3$	...	$2n$

➤ برنامه‌ای که رشته‌ای را خوانده، با استفاده از یک متد فضای خالی سمت چپ (ابتدا رشته) را حذف می‌کند و نمایش می‌دهد (کد اسکی کاراکترهای خالی ۳۲ است).

➤ برنامه‌ای که رشته‌ای را خوانده، با استفاده از اشاره‌گر و بازگشتی، طول رشته را حساب کرده، نمایش می‌دهد.

➤ **پروژه برنامه نویسی ۱:** یک پروژه هواپیمایی قصد دارد سیستم ذخیره و فروش بلیط را کامپیوتری کند. برنامه سیستم رزرو بلیط مراحل زیر را انجام می دهد (تعداد صندلی ۴۰ می باشد که شماره ۱ تا ۲۰ برای سیگاری ها و از ۲۱ تا ۴۰ برای غیر سیگاری ها در نظر گرفته می شود. در ضمن حداکثر ۱۰ شماره پرواز از ۱ تا ۱۰ داریم):

الف: آرایه دوبعدی با ۴۰ سطر و ۱۰ ستون در نظر گرفته شده که ستون های صفر تا ۹ شماره پروازهای ۱ تا ۱۰ را تعیین می کنند.

ب: شماره مسافر، نوع مسافر (سیگاری (۱)، غیر سیگاری (۰)) و شماره پرواز را دریافت می کند، سپس بلیط را رزرو می کند.

ج: اگر نوع مسافر سیگاری باشد، با توجه به شماره پرواز، به آن شمار پرواز در بخش سیگاری صندلی تخصیص می دهد. در صورتی که پر بودن بخش سیگاری ها از مشتری سؤال می کند آیا در بخش غیر سیگاری ها پذیرفته شود یا خیر و برعکس.

د: شماره مسافر به شماره صندلی در آرایه تخصیص می یابد.

سپس کلید n را فشار دهید تا وضعیت صندلی های پرواز را به شکل زیر ببینید:

➤ **پروژه برنامه نویسی ۲:** برنامه ای که منوی نمایش می دهد. از این منو کاربر می تواند یکی از گزینه های زیر را انتخاب کند:

۱ Read Array، اطلاعات آرایه اول را می خواند.

۲ Read Array، اطلاعات آرایه دوم را می خواند.

Find Share، اشتراک بین آرایه اول و دوم را پیدا می کند.

Find Union، اجتماع بین آرایه اول و دوم را پیدا می کند.

Find difference a Form b، تفاضل آرایه a را از b حساب می کند.

Find difference b Form a، تفاضل آرایه b را از a حساب می کند.

➤ **پروژه برنامه نویسی ۳:** برنامه ای که دو آرایه  $3 \times 3$  را خوانده، دو آرایه را با هم جمع، تفریق و ضرب می نماید. در این برنامه امکاناتی برای نمایش دو آرایه دارد. در ضمن این برنامه می تواند N، M و k را بخواند. سپس حاصل عبارت  $N * A - M * B + K$  را محاسبه کنند و نمایش دهد (A و B به ترتیب آرایه های اول و دوم هستند).

➤ **پروژه برنامه نویسی ۴:** برنامه ای که عناصر یک ماتریس سه در سه را خوانده، اعمال زیر را انجام می دهد:

تعیین می کند ماتریس بالا مثلثی است یا نه؟

تعیین می کند ماتریس پایین مثلثی است یا نه؟

تعیین می کند ماتریس بالا و پایین مثلثی است یا نه؟

تعیین می کند که آیا تمام عناصر هر سطر ماتریس برابرند یا نه؟

تعیین می کند که تمام عناصر هر ستون ماتریس برابرند یا نه؟

➤ پروژه برنامه نویسی ۵. فرض کنید بخواهیم وضعیت مدال های بین المللی یک کشور را در رشته ورزشی خاصی در مهر و موم های ۱۳۸۵ تا ۱۳۹۵ مطابق جدول زیر دریافت و پردازش کنیم:

سال	تعداد مدال طلا	تعداد مدال نقره	تعداد مدال برنز
۱۳۸۵			
۱۳۷۱			
۱۳۷۲			
۱۳۹۴			
۱۳۹۵			

برنامه ای که ضمن دریافت داده های لازم (خانه های خالی طول) موارد زیر را نمایش می دهد.

تعداد کل مدال های کسب شده در همه ی سال ها  
تعداد کل مدال های طلای کسب شده در همه ی سال ها  
شماره سال هایی که هیچ مدالی کسب نشده  
سالی که بیش ترین مدال کسب شده  
سالی که بیش ترین مدال طلا کسب شده است.  
امتیاز تمام سال ها به تفکیک سال، با فرض این که هر مدال طلا ۳ امتیاز، هر مدال نقره ۲ امتیاز و هر مدال برنز ۱ امتیاز داشته باشد.  
تعداد سال هایی که امتیاز بیش از ۱۰ کسب شده است.

➤ پروژه برنامه نویسی ۶. در یک تورنمنت ژیمناستیک با شرکت ۱۰ ورزش کار و ۵ داور امتیاز منظر خود را در یک جدولی به شکل زیر یادداشت می کنند. ملاک انتخاب قهرمان امتیازات داده شده توسط داوران به هر ورزشکار است. گزارش گردید یکی از داوران از یک ورزشکار رشوه گرفته تا به قهرمان شدن او کمک کند. اما داور و ورزشکار متخلف شناسایی نشدند. با توجه به این که داور متخلف امتیازات پرت تری نسبت به همکارانش ثبت کرده است، کمیته برگزاری تصمیم گرفته با بررسی داده ها او را شناسایی کرده و امتیازات داده شده توسط وی را از جدول حذف نموده و در نهایت نام قهرمان واقعی و نام ورزشکار رشوه دهنده را مشخص کند. این برنامه ای کار را انجام می دهد.

شماره پیراهن	نام ورزشکار	امتیاز داور	امتیاز داور	.....	امتیاز داور
-----------------	----------------	----------------	----------------	-------	----------------

۶۰۰ برنامه C# با حل آن‌ها ۸۷

پنجم		دوم	اول		
					۱
					۲
					۳
					⋮
					۹
					۱۰

## برنامه‌نویسی شی‌گرا: کلاس‌ها، وراثت و چندریختی

- برنامه‌ای که کلاسی طراحی می‌کند که اطلاعات زیر را دارد:
  ۱. شماره استادی      ۲. نام استاد      ۳. نام خانوادگی استاد
  ۴. ساعات تدریس      ۵. مبلغ پرداختی به ازای هر ساعت تدریس

در این کلاس مبلغ حق‌التدریس استاد را محاسبه می‌کند که به صورت زیر محاسبه می‌شود:

مبلغ حق‌التدریس = ساعت تدریس \* مبلغ پرداختی به ازای هر ساعت تدریس

برنامه‌ای که از این کلاس استفاده می‌کند.
- برنامه‌ای که دو زمان را مقایسه می‌کند. در این برنامه از کلاسی به نام Time استفاده شده است. این کلاس دارای سه فیلد hour (ساعت)، minute (دقیقه)، second (ثانیه)، دو سازنده و یک متد به نام isEqual() است که دو شی از نوع Time را مقایسه می‌کند.
- برنامه‌ای که پیاده‌سازی ایندکسرها را برای ذخیره و بازیابی چند نام نشان می‌دهد (هدف این برنامه آشنایی با Indexer، پیاده‌سازی و تعریف مجدد آن می‌باشد).
- برنامه‌ای که پیاده‌سازی و استفاده از delegate را نشان می‌دهد.
  -
- برنامه‌ای که کلاس مربع را پیاده‌سازی می‌کند. برای این منظور کلاس Square را ایجاد می‌نماید. متغیر خصوصی side را در این کلاس ایجاد می‌کند که توسط متدهای set و get بتوان به آن دسترسی یافت. کلاس باید شامل دو متد سازنده باشد. یکی از سازنده‌ها آرگومان ندارد و متد سازنده دیگر مقدار side را باید به عنوان آرگومان بپذیرد.
- برنامه‌ای که کلاسی به نام SavingAccount ایجاد می‌کند. این کلاس از متغیر استاتیکی به نام annualInterestRate برای ذخیره سازی نرخ سود همه حساب‌ها استفاده می‌کند. هر شی از این کلاس شامل متغیر خصوصی به نام SavingBalance می‌باشد که مقدار پس‌انداز جاری سپرده را در خود نگه می‌دارد. متدی به نام CalculateModifyInterest() دارد که سود ماهانه را با فرمول زیر محاسبه می‌نماید:

$$(\text{SavingBalance} * \text{annualInterestRate})/12$$

اضافه می‌نماید. کلاس متد استاتیکی به نام `SavingBalance` سپس سود را به یک مقدار جدید `annualInterestRate` که به مقدار فیلد `ModifyInterestRate` تخصیص می‌دهد. برنامه‌ای که از این کلاس استفاده می‌کند. در این برنامه دو شیء به نام- ایجاد می‌کند و به هر یک تراز ۲۰۰۰۰۰ دلار و ۳۰۰۰۰۰ دلار نسبت `saver1` و `saver2` می‌دهد. مقدار ۴٪ قرار می‌دهد و سپس سود ماهانه را حساب می‌کند. `annualInterestRate` می‌دهد. مقدار محاسبه کرده، نمایش می‌دهد. سپس مقدار `saver1` و `saver2` را برآورد و تراز را برای ۵٪ قرار می‌دهد و سود ماه بعد را حساب می‌نماید و تراز جدید `annualInterestRate` (است). `savingBalance` را برای هر یک از حساب‌ها نشان می‌دهد (تراز همان فیلد

➤ کلاسی به نام `ChkNum` که دو متد به نام‌های `IsPrime()` و `LeastComFactor()` دارد. متد `IsPrime()` پارامتری به نام `x` را دریافت کرده، تعیین می‌کند اول است یا خیر و متد `LeastComFactor()`، دو پارامتر به نام‌های `a` و `b` را دریافت می‌کند و کوچکترین فاکتور مشترک بین آن‌ها را برمی‌گرداند (کوچکترین فاکتور مشترک، کوچکترین عدد اولی است که به هر دو پارامتر `a` و `b` بخش پذیر باشد).

➤ اگر طول دو ضلع (`B` و `C`) از مثلث و زاویه بین این دو ضلع `Alpha` باشد. می‌توان طول ضلع سوم را با فرمول زیر حساب کرد:

$$A^2 = B^2 + C^2 - 2 * B * C * \text{Cos}(\text{Alpha})$$

( را با متدهای مساحت و `A` را دریافت می‌کند. طول ضلع سوم (`Alpha` و `C`، `B` کلاسی که محیط را حساب کرده، نمایش می‌دهد. مساحت و محیط مثلث به صورت زیر حساب می‌شوند.

$$\sqrt{p(p-a)(p-b)(p-c)} = \text{مساحت} \quad p = \frac{(a+b+c)}{2} \quad \text{محیط}$$

➤ برنامه‌ای که با استفاده از `Delegate` و متدهای متعدد، حرف اول، آخر و کلیه حروف رشته را به حروف بزرگ تبدیل می‌کند.

➤ برنامه‌ای که تعدادی (۴ عدد) را خوانده، به روش صعودی، و نزولی مرتب می‌نماید و نمایش می‌دهد (روش مرتب‌سازی را به عنوان یک پارامتر از نوع `delegate` دریافت می‌کند).

- برنامه‌ای که نقطه شروع و پایان خطی را دریافت می‌کند، طول خط را محاسبه کرده، نمایش می‌دهد.
- برنامه‌ای که مجموع مساحت‌ها و محیط‌های چند مربع را نمایش می‌دهد (در این برنامه تعداد مربع‌ها و اضلاع آن‌ها به صورت تصادفی تولید می‌گردند).
- برنامه‌ای که استفاده از delegate را نشان می‌دهد.
- برنامه‌ای که اندازه پارچه را به یارد دریافت کرده، به متر تبدیل می‌کند (هر متر ۱٫۱۹۶ یارد است).
- برنامه‌ای که قاعده و ارتفاع مثلثی را دریافت کرده، محیط و مساحت مثلث را حساب می‌کند. محیط و مساحت مثلث به صورت زیر محاسبه می‌گردد:  

$$\text{محیط مثلث} = \text{ارتفاع} + \text{قاعده} + \sqrt{(\text{ارتفاع})^2 + (\text{قاعده})^2}$$

$$۲ / \text{ارتفاع} \times \text{قاعده} = \text{مساحت مثلث}$$
- برنامه‌ای که زمان را دریافت کرده و آن را به صورت Am (قبل از ظهر) یا Pm (بعد از ظهر) نمایش می‌دهد. این برنامه کلاس‌های زیر را دارد:
- برنامه‌ای که شماره کارمندی، نام، تاریخ استخدام و حقوق ناخالص تعدادی کارمند را مقدار دهی می‌کند، مالیات، بیمه و حقوق خالص را محاسبه می‌کند. در پایان مجموع حقوق خالص را نمایش می‌دهد. مالیات، بیمه و حقوق خالص به صورت زیر محاسبه می‌گردند.  

$$۱۰۰ / ۲ * \text{حقوق ناخالص} = \text{بیمه}$$

اگر حقوق بیشتر از ۴۸۵۰۰۰۰ ریال باشد، آنگاه مالیات برابر است با

$$۱۰۰ / ۱۰ * \text{حقوق ناخالص} = \text{مالیات}$$

وگرنه، مالیات صفر در نظر گرفته می‌شود.

$$\text{بیمه} - \text{مالیات} - \text{حقوق ناخالص} = \text{حقوق خالص}$$
- برنامه‌ای که شعاع بزرگ (طولی) و شعاع کوچک (عرضی) بیضی را دریافت کرده، مساحت بیضی را با فرمول زیر محاسبه می‌کند:  

$$= \text{مساحت} \text{ } \pi * \text{شعاع کوچک} * \text{شعاع بزرگ}$$

- برنامه‌ای که کلاس‌های زیر را تعریف کرده و از آن‌ها استفاده می‌کند:
- برنامه‌ای که کلاس‌های مربع (Square) و مستطیل را پیاده‌سازی می‌کند. کلاس مستطیل از کلاس مربع مشتق می‌شود.
- برنامه‌ای که کلاس‌های زیر را ایجاد کرده، از آن‌ها استفاده می‌کند:
  - کلاس Shape، کلاسی abstract است که فقط دو متد Area() (برای محاسبه مساحت) و Volume (برای محاسبه حجم) دارد.
  - کلاس Point، مشتق کلاس Shape است که دارای اعضای داده‌ای x و y است.
  - کلاس Circle، مشتق کلاس Point است. علاوه بر این که اعضای کلاس Point را به ارث می‌برد، دارای اعضای زیر می‌باشد:
    - عضو داده‌ای شعاع (radius)، شعاع دایره را نگهداری می‌کند.
    - متد Area()، برای محاسبه مساحت دایره به کار می‌رود.
  - کلاس Cylinder (استوانه)، مشتق کلاس Circle است که علاوه بر اعضای Circle به ارث می‌برد، دارای اعضای زیر نیز است:
    - عضو داده‌ای ارتفاع (height) است.
    - متد Area()، برای محاسبه مساحت استوانه به کار می‌رود. مساحت استوانه به شکل زیر محاسبه می‌شود:  
ارتفاع استوانه \* محیط دایره + ۲ \* مساحت دایره
    - متد Volume()، حجم استوانه را محاسبه کرده، برمی‌گرداند. حجم استوانه برابر است با:  
ارتفاع \* مساحت دایره
- برنامه‌ای که از این کلاس‌ها استفاده می‌کند.
- کلاس استاتیکی که با دو متد درجه سانتی‌گراد را به فارنهایت و درجه فارنهایت را به سانتی‌گراد تبدیل می‌کند. برنامه‌ای که از این کلاس استفاده می‌کند. برای تبدیل درجه‌ها از فرمول‌های زیر استفاده می‌شود.
  - درجه به فارنهایت =  $۳۲ + ۹/۵ * \text{درجه سانتی‌گراد}$
  - درجه به سانتی‌گراد =  $۵/۹ * (\text{درجه فارنهایت} - ۳۲)$
- در این برنامه منویی نمایش داده می‌شود تا نوع درجه را دریافت کند.

➤ برنامه‌ای که کد کالا، نام کالا، تعداد فروش، قیمت را دریافت می‌کند و قیمت فروش را محاسبه کرده، نمایش می‌دهد. در این برنامه از یک واسط به نام IProduct استفاده شده است.

فصل ۶	فایل‌ها
	<p>➤ برنامه‌ای که نام فایلی را خوانده، تعیین می‌کند که آیا فایل یک فایل BMP است یا خیر؟ فایلی با نوع BMP است که بایت اول آن ۰x۴۲ و بایت دوم آن ۰x۴۰ باشد.</p>
	<p>➤ برنامه‌ای که نام، سن و تاریخ تولد فردی را دریافت کرده، در فایل باینری ۱.dat در پوشه Data درایو D می‌نویسد.</p>
	<p>➤ برنامه‌ای که نام یک فایل باینری که اطلاعات آن نام، سن و سال تولد است را می‌خواند و اطلاعات آن را بر روی صفحه نمایش نشان می‌دهد.</p>
	<p>➤ برنامه یک برنامه به زبان C# را دریافت کرده به یک برنامه در زبان جاوا تبدیل می‌کند. کلمات برای این منظور bool، string، Console.WriteLine و bool، Console.Write بترتیب باید به String، System.Out.WriteLine و System.Out.Print تبدیل شوند.</p>
	<p>➤ برنامه‌ای که نام دو فایل متنی را از ورودی خوانده، اطلاعات فایل اول را برعکس (از آخرین خط به اولین خط) در فایل دوم کپی می‌کند.</p>
	<p>➤ برنامه‌ای که نام فایلی را خوانده، تعیین می‌کند که آیا این فایل از نوع GIF است یا خیر؟ فایلی با نوع GIF است که بایت‌های اول تا چهارم آن "GIF۸" باشند.</p>
	<p>➤ برنامه‌ای که نام فایلی را خوانده و اطلاعات آن فایل را به حروف بزرگ تبدیل کرده با پسوند dat با همان نام در فایل دیگری می‌نویسد.</p>
	<p>➤ برنامه‌ای که نام یک فایل باینری را خوانده، حروف کوچک آن را به حروف بزرگ تبدیل کرده، در فایلی با همان نام و پسوند UPP می‌نویسد.</p>
	<p>➤ برنامه‌ای که نام فایل باینری را خوانده و اطلاعات آن را در یک فایل باینری دیگر با همان نام و پسوند Inv می‌نویسد.</p>

- برنامه‌ای که نام یک فایل متنی را خوانده، بایت‌های آن را رمز گذاری می‌کند و در فایلی با پسوند Enc می‌نویسد.
- برنامه‌ای که نام فایل متنی را خوانده، تعداد کلمات آن را شمارش می‌کند. کلمات در فایل با کاراکتر (Blank) از یکدیگر جدا می‌شوند.
- برنامه‌ای که نام یک فایل باینری را خوانده، ابتدا تشخیص می‌دهد این فایل یک فایل BMP است یا خیر (فایلی BMP است که بایت اول و دوم آن به ترتیب 0x42 و 0x4D باشد) و سپس عرض و ابتدا فایل را نمایش می‌دهد (عرض و ارتفاع فایل BMP در بایت ۱۸ به بعد هر یک به صورت یک عدد صحیح قرار دارند).
- برنامه‌ای که نام دو فایل را خوانده و فایل اولی را در فایل دومی کپی می‌کند. در هنگام کپی کردن هر مرحله ۵۱۲ k از فایل اولی می‌خواند و در فایل دومی می‌نویسد.
- برنامه‌ای که نام فایلی را خوانده، سپس اندازه (تعداد بایت) را می‌خواند و فایل اولی را با توجه به اندازه به چند فایل دیگر تقسیم می‌کند (شماره فایل‌ها، همان نام فایل اصلی به همراه شماره‌های ۰۰۱، ۰۰۲، ..... می‌باشد).
- برنامه‌ای که نام یک فایل را خوانده، تشخیص می‌دهد که آیا این فایل BMP است یا خیر (فایلی BMP است که با بیت‌های ۰ و یک آن بترتیب کاراکترهای 'B' و 'M' باشند).
- برنامه‌ای که نام یک فایل CSV را خوانده، کلمات آن را جدا کرده و هر کلمه را در یک سطر چاپ می‌کند. فایل‌های CVS، فایل‌هایی متنی هستند که کلمات آن‌ها با کاما از یکدیگر جدا می‌شوند.
- برنامه‌ای که نام دو فایل را می‌خواند و تعداد کاراکترهایی از فایل اول و دوم که نظیر به نظیر با هم برابرند را نمایش می‌دهد.
- برنامه‌ای که نام فایلی را خوانده و کاراکترهای قابل چاپ آن را در فایلی با همان نام به علاوه "۰۱. txt" می‌نویسد.

## ۶۰۰ برنامه C# با حل آن‌ها ۹۵

- برنامه‌ای که نام یک فایل را خوانده، اطلاعات آن را به صورت متنی و مکان آن را به صورت هگزا دیسمال (مبنای ۱۶) به صورت ۲۲ خط ۲۲ خط در هر صفحه نمایش می‌دهد.
- برنامه‌ای که نام فایلی را به صورت آرگومان یا از ورودی می‌خواند و آن فایل را از دیسک حذف می‌کند.
- نامه‌ای که تعدادی رشته را خوانده و سپس  $n$  را می‌خواند و  $n$  خط را خوانده، خطوطی که رشته اول خوانده شده در آن‌ها نباشد، در فایلی که نام آن را از ورودی می‌خواند، می‌نویسد و در پایان اطلاعات فایل را نمایش می‌دهد.
- برنامه‌ای که نام فایلی را خوانده، سپس  $n$  را می‌خواند و  $n$  سطر آخر فایل را از ابتدا به انتها نمایش می‌دهد.
- برنامه‌ای که نام یک فایل و عدد  $n$  را خوانده، سپس  $n$  سطر انتهای فایل را از انتها به ابتدا نمایش می‌دهد.
- برنامه‌ای که نام یک فایل و  $n$  را خوانده،  $n$  سطر ابتدای فایل را از ابتدا به انتها نمایش می‌دهد.
- برنامه‌ای که نام یک فایل و عدد  $n$  را خوانده،  $n$  سطر ابتدای فایل را از انتها به ابتدا نمایش می‌دهد.
- برنامه‌ای که نام فایلی را خوانده، اطلاعات شماره سطرهای فرد آن را نمایش می‌دهد.
- برنامه‌ای که نام یک فایل و عدد  $n$  را خوانده و از  $n$  امین سطر فایل تا انتهای فایل را نمایش می‌دهد.
- برنامه‌ای که نام فایلی را خوانده، تعداد خطوط فایل را نمایش می‌دهد.

- برنامه‌ای که نام دو فایل و یک کاراکتر را خوانده و از فایل اول کاراکتر خوانده شده را حذف کرده و نتیجه را در فایل دوم می‌نویسد.
- برنامه‌ای که نام چند فایل را از ورودی می‌خواند و اطلاعات آن فایل‌ها را در فایل kol.txt در پوشه Data درایو D ادغام می‌کند (برای خاتمه کاربر باید به جای نام فایل Enter را فشار دهد (یعنی نام فایل را وارد نکند)).
- برنامه‌ای که نام سه فایل را از کاربر می‌خواند و خطوط فایل‌های اول و دوم را یک دو میان در فایل سوم می‌نویسد (یعنی، یک خط از فایل اول و یک خط از فایل دوم و همین روند را ادامه می‌دهد).
- برنامه‌ای که نام یک فایل را خوانده و اطلاعات فایل اول را بطور معکوس (از آخرین خط به اولین خط) در فایل newFile.txt در پوشه Data درایو D می‌نویسد.
- برنامه‌ای که نام دو فایل را خوانده، سپس دو عدد را می‌خواند و اطلاعات فایل اول را در فایل دوم می‌نویسد، بطوریکه خطوط بین بازه بسته بین این دو عدد را در فایل دوم نمی‌نویسد.
- برنامه‌ای که نام فایلی را خوانده، بخش توسعه (پسوند فایل) را نمایش می‌دهد.
- برنامه‌ای که نام دو فایل را خوانده و از فایل اول تمام خطوط جدید (کاراکتر '\n') را حذف کرده و در فایل دوم می‌نویسد.
- برنامه‌ای که نام فایلی را خوانده، تعداد خطوط، تعداد کلمات، تعداد کاراکترها و تعداد کاراکترهای سفید آن را شمارش کرده و نمایش می‌دهد.
- برنامه‌ای که نام فایلی را خوانده، تعداد حروف کوچک، تعداد حروف بزرگ، تعداد فضای سفید، تعداد ارقام، تعداد کاراکترهای کنترل و تعداد نمادهای آن را نمایش می‌دهد.
- برنامه‌ای که نام فایلی را خوانده، سپس یک کلمه را می‌خواند و تعداد تکرار این کلمه در فایل را شمارش می‌کند.

## ۶۰۰ برنامه C# با حل آن‌ها ۹۷

- برنامه‌ای که نام فایلی را خوانده، حرف اول، وسط و آخر هر سطر را نمایش می‌دهد.
- برنامه‌ای که نام فایل‌ها و تعدادی عدد ( $n$ ) را خوانده،  $n$  کاراکتر ابتدای فایل‌ها را نمایش می‌دهد. این برنامه موقعی خاتمه می‌یابد که  $n$  عدد منفی وارد شود یا در نام فایل چیزی وارد نگردد.
- برنامه‌ای که نام فایلی را خوانده، اطلاعات آن فایل از قبیل نام کامل، پسوند فایل، نام دایرکتوری، را نمایش می‌دهد.

- برنامه‌ای که نرخ حقوق به ازای هر ساعت و تعداد ساعت کارکرد یک کارمند را خوانده، میزان حقوق او را محاسبه می‌کند و نمایش می‌دهد (از حقوق ۱۰ درصد به‌عنوان مالیات کسر می‌شود. حقوق خالص برابر با حقوق منهای مالیات است).
- برنامه‌ای که مختصات دوتنقطه را دریافت کرده، فاصله بین آن دوتنقطه را محاسبه می‌نماید و نمایش می‌دهد.
 
$$\text{فاصله بین دو نقطه} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$
- زمینی به ابعاد  $16 \times 13$  وجود دارد. موزاییک‌هایی به ابعاد  $3 \times 2$ ، برای فرش کردن زمین نیاز است. اولاً تعیین کنید چند موزاییک نیاز است. آیا قسمتی از زمین می‌ماند که با موزاییک فرش نشود؟
- برنامه‌ای که نقطه تقاطع دو خط  $y = 3x + c$  و  $y = ax + b$  را نمایش می‌دهد (برنامه a، b و c را از ورودی می‌خواند). مختصات نقطه تقاطع به‌صورت زیر به دست می‌آید:
 
$$x = (b - c) / (3 - a) \quad y = 3 * x + c$$
- برنامه‌ای که a و b را خوانده، مقدار  $y = b * \sin(a)$  را نمایش می‌دهد.
- برنامه‌ای که Four را نمایش داده، چهار بار بوق کامپیوتر را به صدا در می‌آورد، سپس به خط بعد می‌رود، Five را نمایش داده، ۵ بار بوق کامپیوتر را به صدا در می‌آورد.
- برنامه‌ای که عدد یک و یک کاراکتر قلب، سپس عدد ۲، دو کاراکتر قلب و عدد ۳، سه کاراکتر قلب، ۴ و کاراکتر قلب را نمایش می‌دهد.
- برنامه‌ای که تعداد بایت‌هایی را نمایش می‌دهد که متغیرهای با انواع long، int، char، double و float اشغال می‌کنند.
- برنامه‌ای که یک کاراکتر را خوانده، کد اسکی آن را نمایش می‌دهد (در این برنامه متغیر ch کاراکتر خوانده‌شده را نگه‌داری می‌کند).
- برنامه‌ای که عددی را خوانده، با استفاده از عملگر یا انحصاری تشخیص می‌دهد، عدد خوانده‌شده زوج است یا فرد؟

## ۶۰۰ برنامه C# با حل آن‌ها ۹۹

- برنامه‌ای که ابتدا عددی را خوانده، سپس  $n$  را می‌خواند و مقدار  $n$  امین بیت عدد اول خوانده شده را نمایش می‌دهد.
- برنامه‌ای که ابتدا عددی را خواند، سپس  $n$  را می‌خواند و  $n$  امین بیت عدد را برابر ۱ (یک) قرار می‌دهد.
- برنامه‌ای که ابتدا عددی را خوانده، سپس  $n$  را می‌خواند و  $n$  امین بیت عدد خوانده شده را صفر می‌نماید. به عنوان مثال، شکل زیر ۴ امین بیت عدد ۵۵ را صفر می‌نماید.

55	←	num(in decimal)
4	←	n(in decimal)
00000001	←	1(in binary)
00010000	←	$1 \ll n$
11101111	←	$\sim(1 \ll n)$
00110111	←	num(in binary)
00100111	←	Num & ( $\sim(1 \ll n)$ )

- برنامه‌ای که ابتدا عددی را خوانده، سپس عدد  $n$  را می‌خواند و  $n$  امین بیت عدد خوانده شده را معکوس می‌کند. یعنی، چنانچه این بیت یک باشد، به صفر و اگر صفر باشد، به یک تبدیل می‌نماید.

- برنامه‌ای که دو عدد را خوانده، با استفاده از عملگر یا انحصاری بیتی آن‌ها را باهم تعویض می‌نماید و نمایش می‌دهد. مراحل انجام این کار در شکل زیر آمده است:

a	00011010	b	01000001	a	00011010
b ^	01000001	x ^	01011011	x ^	01011011
<hr/>					
x	01011011	a	00011010	b	01000001

- برنامه‌ای که سه عدد را به عنوان قاعده کوچک، قاعده بزرگ و ارتفاع یک دوزنقه خوانده، مساحت دوزنقه را نمایش می‌دهد. مساحت دوزنقه برابر با:  

$$\left( \text{قاعده بزرگ} + \text{قاعده کوچک} \right) * \text{ارتفاع} * \frac{1}{2}$$

➤ برنامه‌ای که عددی را خوانده، تمام بیت‌های عدد خوانده شده را معکوس می‌کند.  
یعنی، چنانچه بیت یک باشد به صفر و اگر صفر باشد، به یک تبدیل می‌نماید.

➤ برنامه‌ای که عددی را خوانده، تعیین می‌کند که مربع کامل است یا خیر؟ عددی مربع کامل است که مجموع تعدادی اعداد فرد پشت سر هم از ۱ و بعد آن برابر آن عدد شود. به عنوان مثال، اعداد زیر مربع کامل هستند:

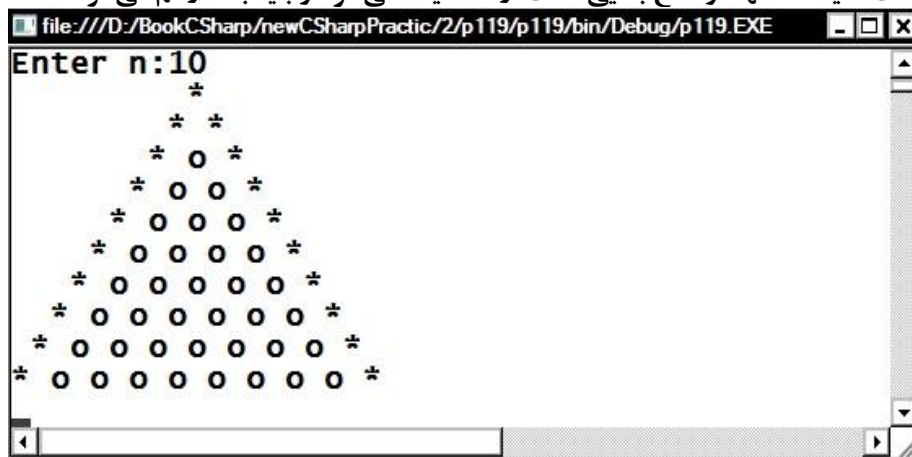
$$\begin{aligned}\sqrt{9} &= 1 + 3 + 5 & \sqrt{1} &= 1 \\ \sqrt{16} &= 1 + 3 + 5 + 7 & \sqrt{4} &= 1 + 3\end{aligned}$$

➤ برنامه‌ای که  $n$  و  $x$  را خوانده، حاصل  $n$  جمله سری زیر را نمایش می‌دهد:

$$\frac{x + \frac{x}{2}}{1!} + \frac{x^3 - \frac{x}{4}}{2!} + \frac{x^5 + \frac{x}{8}}{3!} + \dots$$

➤ برنامه‌ای که عددی را خوانده، ارقام زوج آن را نمایش می‌دهد. به عنوان مثال، برای عدد ۱۳۵۲۷۴، عدد ۲۴ را نمایش خواهد داد.

➤ برنامه‌ای که  $n$  را خوانده، یک مثلث به شکل زیر با  $n$  سطر در خروجی نمایش می‌دهد. دقت کنید که تنها دو ضلع بالایی مثلث از \* کشیده می‌شود و بقیه با O رسم می‌گردند.



➤ ۱۲۰. برنامه‌ای که یک عدد را خوانده، در خروجی عددی را نمایش دهد که هر رقم آن یک واحد اضافه شده است. به عنوان مثال، اگر کاربر عدد ۴۵۹۳۴ را وارد کند، در خروجی ۵۶۰۴۵ چاپ شود (رقم ۹ به صفر تبدیل گردد).

➤ ۱۲۱. برای محاسبه  $\pi$  می‌توان از رابطه زیر استفاده کرد:

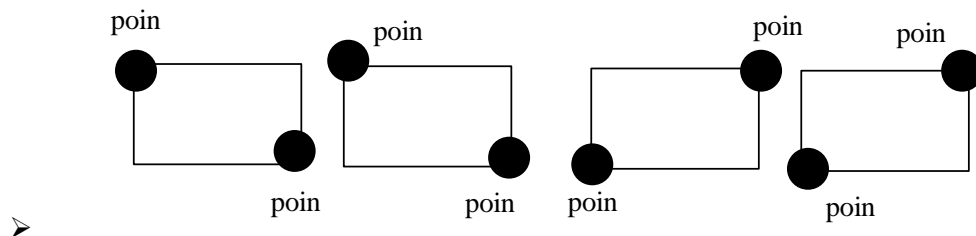
$$\frac{\pi}{2} = \frac{2}{1} + \frac{2}{3} + \frac{4}{3} + \frac{4}{5} + \frac{6}{5} + \frac{6}{7} + \dots$$

➤ برنامه‌ای که تعداد جملات سری فوق (n) را به عنوان یک عدد صحیح دریافت کرده و مقدار  $\pi$  را با استفاده از این رابطه محاسبه کرده و نمایش می‌دهد.

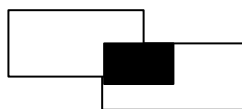
➤ برنامه‌ای که مختصات سه نقطه را از ورودی می‌گیرد و تعیین می‌کند که آن سه نقطه هم خط هستند یا خیر؟

➤ اگر سه نقطه هم خط بودند، خروجی برنامه، یک معادله به صورت  $ax + by + c = 0$  خواهد بود، وگرنه خروجی "No" خواهد بود. ورودی به صورت شش عدد اعشاری خواهد بود که مختصات سه نقطه A و B و C را بیان می‌کند. سه نقطه  $A(x_1, y_1)$ ،  $B(x_2, y_2)$  و  $C(x_3, y_3)$  را هم خط گویند، هرگاه شیب دو خط AB و BC باهم برابر باشند. شیب دو خط AB با X فرمول  $m = \frac{y_2 - y_1}{x_2 - x_1}$  قابل محاسبه است. همچنین، معادله خط AB به صورت  $y - y_1 = m(x - x_1)$  نوشته می‌شود.

➤ برنامه‌ای که مختصات دو مستطیل را از ورودی گرفته و مساحت ناحیه‌ی مشترک بین دو مستطیل را حساب می‌نماید. به ازاء هر مستطیل دو نقطه (جمعاً چهار عدد اعشاری) باید گرفته شود. توجه کنید که هر مستطیل با استفاده از دو نقطه می‌تواند به یکی از چهار فرم زیر نمایش داده شود.



➤



➤ به عنوان نمونه، هدف این سؤال پیدا کردن ناحیه پررنگ بین دو مستطیل زیر می‌باشد. البته این احتمال وجود دارد که ناحیه مشترک وجود نداشته باشد که در این صورت مساحت ناحیه مشترک صفر خواهد بود.

➤ برای هر کسر به صورت  $1/k$  (k عدد صحیح بزرگ‌تر از صفر و کوچک‌تر از ۱۰۰۰)، دو

$$\frac{1}{k} = \frac{1}{x} + \frac{1}{y}$$

عدد صحیح x و y وجود دارد که عبارت مقابل برقرار باشد:

### ۶۰۰ برنامه C# با حل آن ها ۱۰۳

برنامه‌ای که مقدار  $k$  را از دریافت می‌کند و تمام  $x$  و  $y$ هایی که در این عبارت صدق می‌کنند را نمایش می‌دهد. توجه کنید اگر  $k$  در بازه درست نباشد، برنامه باید در خروجی هشدار خطا دهد.  
نمونه ورودی: ۱۲ خروجی

$$\begin{array}{ccc} \frac{1}{12} = \frac{1}{13} + \frac{1}{156} & \frac{1}{12} = \frac{1}{14} + \frac{1}{84} & \frac{1}{12} = \frac{1}{15} + \frac{1}{60} \\ \frac{1}{12} = \frac{1}{16} + \frac{1}{48} & \frac{1}{12} = \frac{1}{18} + \frac{1}{36} & \frac{1}{12} = \frac{1}{20} + \frac{1}{30} \\ \frac{1}{12} = \frac{1}{24} + \frac{1}{24} & \frac{1}{12} = \frac{1}{21} + \frac{1}{28} & \end{array}$$

➤ برنامه‌ای که به روش تنصیف معادله زیر را تا چهار رقم اعشار حل می‌کند.

$$x + e^x = 5 \quad F(x) = x + e^x - 5 = 0$$

برای حل این معادله دو تخمین اولیه نیاز داریم. یکی عددی که به ازای آن مقدار معادله مثبت باشد ( $x_p$ ) و دیگری که مقدار منفی به معادله بدهد ( $x_n$ ). تخمین جدید در هر مرحله ( $x_m$ ) میانگین دو تخمین قبل است.

➤ برنامه‌ای که  $n$  را خوانده، حاصل  $n$  جمله سری زیر را نمایش می‌دهد.

$$1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots$$

➤ برنامه‌ای که همه حروف الفبای انگلیسی را به ترتیب چاپ می‌کند به‌طوری که هر حرف بزرگ، مشابه کوچک آن هم نمایش داده شود.

➤ برنامه‌ای که تعدادی عدد که به‌صورت مرتب‌شده صعودی وارد می‌شوند، را از ورودی خوانده، حاصل جمع آن‌ها را نمایش می‌دهد. اگر عدد بعدی از عدد قبلی کمتر وارد شود، برنامه خاتمه می‌یابد.

➤ ۱۲۹. برنامه‌ای که دو عدد را خوانده و حاصل عبارت زیر را در بازه بین این دو عدد محاسبه کرده، نمایش می‌دهد.

$t$  بین دو عدد است. اگر  $t$  کوچک‌تر از صفر باشد، حاصل عبارت

$$y = 3 * t * t + 5;$$

وگرنه،

$$y = -3 * t * t + 5;$$

را محاسبه می‌کند.

➤ برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل  $n$  جمله عبارت زیر را نمایش می‌دهد:

$$\frac{-x^1}{1+2} + \frac{x^3}{3+4} - \frac{x^5}{5+6} + \dots$$

➤ برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل  $n$  جمله عبارت زیر را حساب می‌کند:

$$-\frac{x^2}{1*3} + \frac{x^4}{3*5} - \frac{x^6}{5*7} + \dots$$

➤ برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل  $n$  جمله سری زیر را محاسبه کرده، نمایش می‌دهد.

$$X/3 + x^2/(2*3^2) + x^3/(3*3^3) + \dots$$

➤ برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل  $n$  جمله سری زیر را محاسبه کرده و نمایش می‌دهد:

$$\frac{2x^1}{1+2} - \frac{4x^2}{2+4} + \frac{8x^3}{3+8} - \dots$$

➤ برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل  $n$  جمله سری زیر را محاسبه کرده، نمایش می‌دهد:

$$x + \frac{x^2}{5} + \frac{x^3}{25} + \frac{x^4}{125} + \dots$$

➤ برنامه‌ای که ابتدا  $n$  را خوانده، سپس به تعداد  $n$  نقطه را به صورت اعشاری ( $x, y$ ) می‌خواند و معادله  $y = mx + b$  را به دست آورد.  $m$  و  $b$  صورت‌های زیر محاسبه می‌شوند:

$$m = \frac{(\sum xy) - \bar{y}(\sum x)}{(\sum xx) - \bar{x}(\sum x)} \quad b = \bar{y} - m\bar{x}$$

$\bar{X}$  و  $\bar{Y}$  به ترتیب میانگین  $X$  و  $Y$  هستند. در این برنامه متغیرهای  $n$  (تعداد نقاط)،  $X$  و  $Y$  (مختصات هر نقطه در هر مرحله)،  $\sum X$  (مجموع مقادیر  $x$ )،  $\sum Y$  (مجموع مقادیر  $y$ )،  $\sum XY$  (مجموع  $x*y$ )،  $\text{mean}X$  (میانگین  $x$ )،  $\text{mean}Y$  (میانگین  $y$ )،  $m$  (ضریب خط)،  $b$  (مقدار ثابت خط) و  $i$  (شمارنده از ۱ تا  $n$ ) را داریم.

➤ برنامه‌ای که تعداد سطرها را خوانده و حاصل خروجی زیر را تولید می‌کند (برای تعداد سرهای ۶ خروجی زیر را نمایش می‌دهد):

1					
2	3				
4	5	6			
7	8	9	10		
11	12	13	14	15	
16	17	18	19	20	21

➤ برنامه‌ای که تعداد سطرها را خوانده و خروجی را با الگوی زیر نمایش می‌دهد

## ۶۰۰ برنامه C# با حل آن‌ها ۱۰۵

```
file:///D:/BookCSharp/newCSharpPractic/2/p137/p137/bin/Debug/p137.EXE
Input number of rows : 7
  1
 2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
22 23 24 25 26 27 28
```

➤ برنامه‌ای که تعداد سطرها را خوانده و الگوی زیر را با ستاره (\*) چاپ می‌کند:

```
file:///D:/BookCSharp/newCSharpPractic/2/p138/p138/bin/Debug/p138.EXE
Input number of rows : 10
  *
 * *
* * *
 * * * *
  * * * * *
   * * * * *
    * * * * *
     * * * * *
      * * * * *
       * * * * *
```

➤ برنامه‌ای که تعداد سطرها را خوانده و الگوی زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p139/p139/bin/Debug/p139.EXE
Input number of rows : 8
  1
 2 2
 3 3 3
 4 4 4 4
 5 5 5 5 5
 6 6 6 6 6 6
 7 7 7 7 7 7 7
 8 8 8 8 8 8 8 8
```

➤ برنامه‌ای که n و k را از ورودی خوانده و سپس n جمله سری زیر را چاپ می‌کند:

➤ 
$$K + KK + KKK + \dots + KKK \dots K$$
  
K عددی بین ۱ تا ۹ است. در پایان، مجموع این سری را نمایش می‌دهد.

➤ برنامه‌ای که x و n را خوانده، حاصل عبارت زیر را محاسبه می‌کند:

➤ 
$$\frac{x^n n^x x^{(n+n)} n^{(x+x)}}{x! n! (x+n)!} \sum_{i=1}^n \frac{x^n n^x (x+n)^i}{(x+2i)! (n+2i)!}$$

➤ . برنامه‌ای که تعیین می‌کند کم‌ارزش‌ترین بیت ۱ است یا ۰ (کم‌ارزش‌ترین بیت، سمت راست‌ترین بیت می‌باشد). به‌عنوان مثال، برای عدد ۱۵ (۰۰۰۰۱۱۱۱)، سمت راست‌ترین بیت ۱ است، ولی برای عدد ۱۲ (۰۰۰۰۱۱۰۰) سمت راست‌ترین بیت ۰ است. اگر این اعداد را با ۱ (یک)، و بیتی نماییم، چنانچه نتیجه صفر شود، سمت راست‌ترین بیت صفر، ولی اگر نتیجه یک شود سمت راست‌ترین بیت یک است:

➤ . برنامه‌ای که تعیین می‌کند باارزش‌ترین بیت یک عدد یک (۱) است یا صفر (۰)؟ باارزش‌ترین بیت عدد، سمت چپ‌ترین بیت آن است.

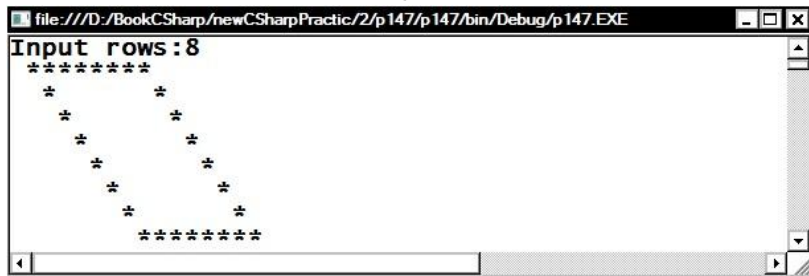
➤ برنامه‌ای که تعداد سطرها (rows) را خوانده و خروجی مانند زیر را نمایش می‌دهد (rows=۹).

➤ ۱۴۵. برنامه‌ای که تعداد سطرها (rows) را خوانده، خروجی زیر را نمایش می‌دهد:

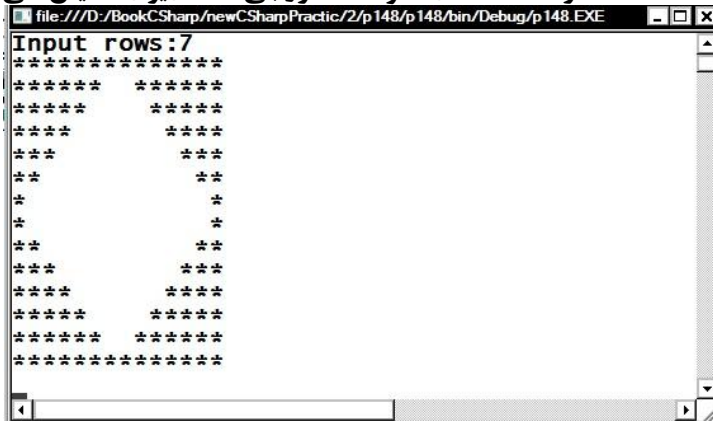
➤ . برنامه‌ای که تعداد سطرها (rows) را خوانده، خروجی مانند زیر را نمایش می‌دهد :

## ۶۰۰ برنامه C# با حل آن‌ها ۱۰۷

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده و خروجی زیر را نمایش می‌دهد:



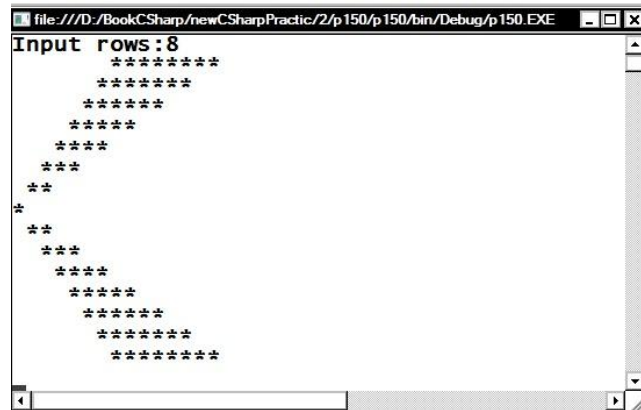
➤ برنامه‌ای که تعداد سطرها (rows) را خوانده، خروجی مانند زیر را نمایش می‌دهد:



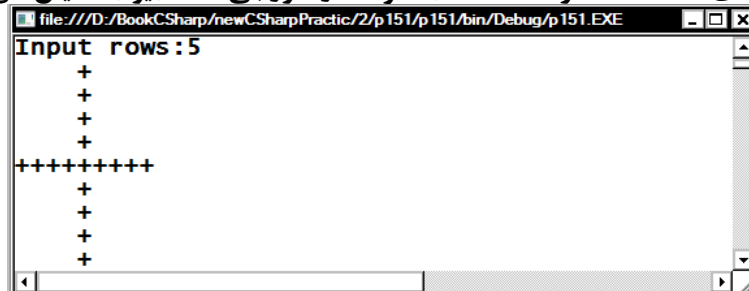
➤ برنامه‌ای که تعداد سطرها (rows) را خوانده، خروجی مانند زیر را نمایش می‌دهد:



➤ برنامه‌ای که تعداد سطرها (rows) را خوانده و خروجی مانند زیر را نمایش می‌دهد:



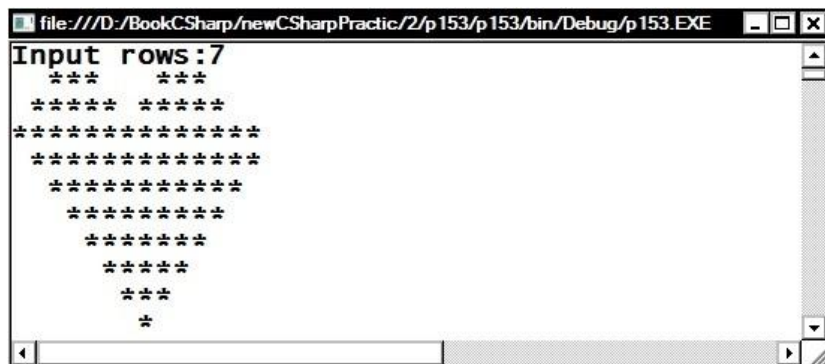
➤ برنامه‌ای که تعداد سطرها (rows) را خوانده و خروجی مانند زیر را نمایش می‌دهد:



➤ برنامه‌ای که تعداد سطرها را خوانده (rows) و خروجی مانند زیر را نمایش می‌دهد:



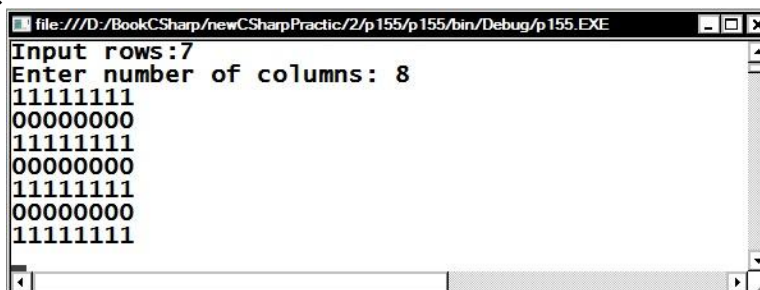
➤ برنامه‌ای که تعداد سطرها (rows) را خوانده، خروجی مانند زیر را نمایش می‌دهد:



➤ برنامه‌ای که تعداد سطرها (rows) و یک نام را خوانده، خروجی مانند زیر را نمایش می‌دهد:



➤ برنامه‌ای که تعداد سطرها (rows) و ستون‌ها (cols) را خوانده، خروجی مانند زیر را نمایش



➤ برنامه‌ای که تعداد سطرها (rows) و تعداد ستون‌ها (cols) را خوانده و خروجی نظیر زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p156/p156/bin/Debug/p156.EXE
Input rows:6
Enter number of columns: 10
0101010101
0101010101
0101010101
0101010101
0101010101
0101010101
0101010101
0101010101
```

➤ برنامه‌ای که تعداد سطرها (rows) و تعداد ستون‌ها (cols) را از ورودی خوانده، خروجی نظیر زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p157/p157/bin/Debug/p157.EXE
Input rows:6
Enter number of columns: 10
1111111111
1000000001
1000000001
1000000001
1000000001
1111111111
```

➤ برنامه‌ای که تعداد سطرها (rows) و تعداد ستون‌ها (cols) را خوانده و خروجی مانند زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p158/p158/bin/Debug/p158.EXE
Input rows:6
Enter number of columns: 12
101010101010
010101010101
101010101010
010101010101
101010101010
010101010101
```

➤ برنامه‌ای که تعداد سطرها (rows) و تعداد ستون‌ها (cols) را خوانده و خروجی نظیر زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p159/p159/bin/Debug/p159.EXE
Input rows:7
Enter number of columns: 9
111101111
111101111
111101111
000000000
111101111
111101111
111101111
```

➤ برنامه‌ای که تعداد سطرها (rows) و تعداد ستون‌ها (cols) را خوانده و خروجی نظیر زیر را نمایش می‌دهد:

### ۶۰۰ برنامه C# با حل آن‌ها ۱۱۱

```
file:///D:/BookCSharp/newCSharpPractic/2/p160/p160/bin/Debug/p160.EXE
Input rows:7
Enter number of columns: 15
1000000000000001
0100000000000010
0010000000000100
0001000000001000
0000100000010000
0000010001000000
0000001010000000
```

➤ برنامه‌ای که تعداد سطرها (rows) و تعداد ستون‌ها (cols) را خوانده، خروجی مانند زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p161/p161/bin/Debug/p161.EXE
Input rows:8
Enter number of columns: 9
123456789
234567899
345678999
456789999
567899999
678999999
789999999
899999999
```

➤ برنامه‌ای که با خواندن تعداد سطرها (rows) و تعداد ستون‌ها (cols) خروجی نظیر زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p162/p162/bin/Debug/p162.EXE
Input rows:5
Input columns: 7
1234567
2345671
3456721
4567321
5674321
```

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده، خروجی مانند الگوی زیر را نمایش می‌دهد"

```
file:///D:/BookCSharp/newCSharpPractic/2/p 163/p 163/bin/Debug/p 163.EXE
Input rows:8
8888888888888888
8777777777777778
8766666666666678
87655555555555678
8765444444445678
8765433333345678
876543222345678
876543212345678
876543222345678
8765433333345678
876544444445678
876555555555678
876666666666678
8777777777777778
8888888888888888
```

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده و الگوی نظیر خروجی زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p 164/p 164/bin/Debug/p 164.EXE
Input rows:6
1
23
4567
89123456
7891234567891234
56789123456789123456789123456789
```

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده و الگوی مانند خروجی زیر را نمایش می‌دهد" در این

```
file:///D:/BookCSharp/newCSharpPractic/2/p 165/p 165/bin/Debug/p 165.EXE
Input rows:8
1          1
12         21
123        321
1234       4321
12345      54321
123456     654321
1234567    7654321
1234567887654321
```

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده، الگوی نظیر زیر را چاپ می‌نماید:

```
file:///D:/BookCSharp/newCSharpPractic/2/p 166/p 166/bin/Debug/p 166.EXE
Input rows:7
1
2 8
3 9 14
4 10 15 19
5 11 16 20 23
6 12 17 21 24 26
7 13 18 22 25 27 28
```

## ۶۰۰ برنامه C# با حل آن‌ها ۱۱۳

➤ برنامه‌ای که تعداد سطرها را خوانده (rows) و الگوی مانند خروجی زیر را نمایش

می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p 167/p 167/bin/Debug/p 167.EXE
Input rows:8
1
2 4
7 11 16
22 29 37 46
56 67 79 92 106
121 137 154 172 191 211
232 254 277 301 326 352 379
407 436 466 497 529 562 596 631
```

➤ برنامه‌ای که تعداد سطرها را خوانده، الگوی نظیر زیر را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p 168/p 168/bin/Debug/p 168.EXE
Input rows:7
1
3 2
4 5 6
10 9 8 7
11 12 13 14 15
21 20 19 18 17 16
22 23 24 25 26 27 28
```

➤ برنامه‌ای که تعداد سطرها (rows) را خوانده، الگوی نظیر خروجی زیر را نمایش

می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p 169/p 169/bin/Debug/p 169.EXE
Input rows:7
1
12
123
4444
33333
222222
1111111
```

➤ برنامه‌ای که عددی را خوانده و بر اساس عدد خوانده‌شده، خروجی طبق الگوی زیر

را نمایش می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p 170/p 170/bin/Debug/p 170.EXE
Enter n:1236754
1236754
123675
12367
1236
123
12
1
```

➤ برنامه‌ای که اگر عدد 24165 را از ورودی بخواند، خروجی با الگوی زیر را نمایش

می‌دهد:

```
file:///D:/BookCSharp/newCSharpPractic/2/p171/p171/bin/Debug/p171.EXE
Enter a number:4525262
4525262
525262
25262
5262
262
62
2
```

➤ برنامه‌ای که ابتدا n را خوانده و سپس الگوی مانند خروجی زیر را نمایش می‌دهد

```
file:///D:/BookCSharp/newCSharpPractic/2/p172/p172/bin/Debug/p172.EXE
Enter n:8
1
12
123
1234
12345
123456
1234567
12345678
1234567
123456
12345
1234
123
12
1
```

➤ برنامه‌ای که n را خوانده و الگوی خروجی نظیر زیر را نمایش می‌دهد

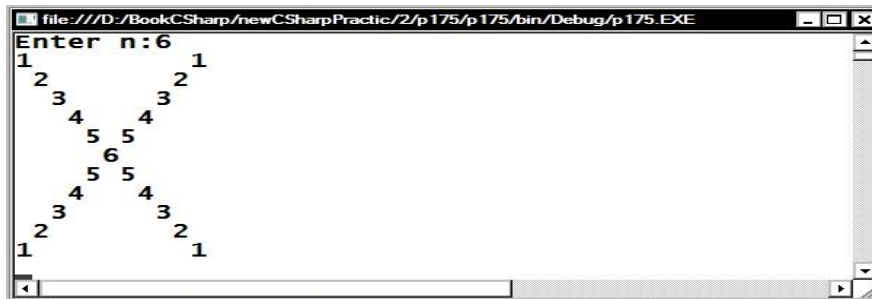
```
file:///D:/BookCSharp/newCSharpPractic/2/p173/p173/bin/Debug/p173.EXE
Enter n:4
1
123
12345
1234567
12345
123
1
```

➤ برنامه‌ای که n را خوانده و الگوی مانند خروجی زیر را نمایش می‌دهد

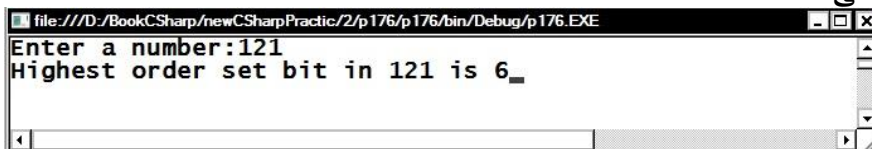
```
file:///D:/BookCSharp/newCSharpPractic/2/p174/p174/bin/Debug/p174.EXE
Enter n:5
*1*
*123*
*12345*
*1234567*
*123456789*
*1234567*
*12345*
*123*
*1*
```

➤ برنامه‌ای که ابتدا n را خوانده و الگوی نظیر خروجی زیر را نمایش می‌دهد:

## ۶۰۰ برنامه C# با حل آن‌ها ۱۱۵



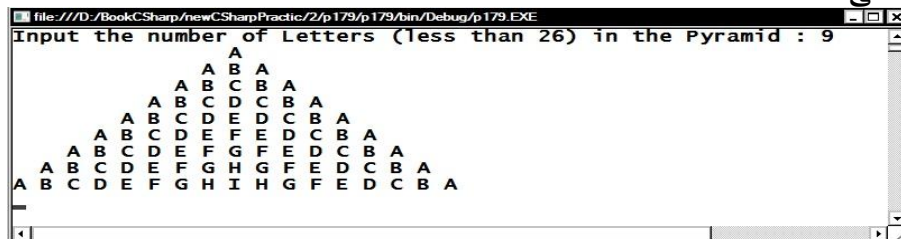
➤ برنامه‌ای که عددی را خوانده، مکان باارزش‌ترین بیت آن را برمی‌گرداند و نمایش می‌دهد.



➤ برنامه‌ای که عددی را خوانده، مکان سمت راست‌ترین بیتی که ۱ باشد را نمایش می‌دهد. به عنوان مثال، اگر عدد ۲۲ در n بیت نمایش داده شود (یعنی ۰۰۰۱۰۱۱۰) مکان سمت راست‌ترین بیت یک آن، ۱ است.

➤ برنامه‌ای که عددی را خوانده، تعداد بیت‌های صفر (۰) و یک (۱) را شمارش می‌کند.

➤ برنامه‌ای که تعداد حروف الفبا را خوانده و الگوی نظیر خروجی زیر را نمایش می‌دهد:



➤ برنامه‌ای که عددی را خوانده، تشخیص می‌دهد عددی زشت (ugly) است یا خیر؟ عددی زشت است که عامل‌های اول آن اعداد ۲، ۳ یا ۵ باشند. برخی از اعداد زشت عبارت‌اند از:

۱ ۲ ۳ ۵ ۶ ۸ ۹ ۱۰ ۱۲.....

➤ برنامه‌ای که ابتدا n را خوانده، سپس n عدد را خوانده و تشخیص می‌دهد که این اعداد تشکیل تصاعد حسابی را می‌دهند یا خیر.

دو سری از اعداد که تشکیل تصاعد حسابی را می‌دهند در زیر آمده‌اند:

➤ -۱ ۳ ۷ ۱۱ ۱۵ ۱۹ .....

➤ ۵ ۱۰ ۱۵ ۲۰ ۲۵ ۳۰ .....

- برنامه‌ای که ابتدا  $n$  را خوانده، سپس  $n$  عدد را می‌خواند. و تشخیص می‌دهد این اعداد تشکیل تصاعد هندسی را می‌دهند یا خیر.  
سری‌های زیر تشکیل تصاعد هندسی را می‌دهند:

۶      ۱۸      ۵۴

○ نرخ ۳.....

۵      ۲,۵      ۱,۲۵

▪ نرخ  $\frac{1}{2}$  .....  
۱۰

- برنامه‌ای که عددی را خواند، سپس اعدادی بین ۱ تا  $n$  را نمایش می‌دهد. که مجموع مکعبات دو یا بیش‌تر عدد برابر با این عدد باشد را نمایش می‌دهد. (نمونه‌ای از این اعداد در شکل زیر آمده است):



$$\begin{aligned} 1729 &= 1^3 + 12^3 = 9^3 + 10^3 \\ 4104 &= 2^3 + 16^3 = 9^3 + 15^3 \\ 13832 &= 2^3 + 24^3 = 18^3 + 20^3 \\ 39312 &= 2^3 + 34^3 = 15^3 + 33^3 \\ 46683 &= 3^3 + 36^3 = 27^3 + 30^3 \\ 32832 &= 4^3 + 32^3 = 18^3 + 30^3 \\ 40033 &= 9^3 + 34^3 = 16^3 + 33^3 \\ 20683 &= 10^3 + 27^3 = 19^3 + 24^3 \\ 65728 &= 12^3 + 40^3 = 31^3 + 33^3 \\ 64232 &= 17^3 + 39^3 = 26^3 + 36^3 \end{aligned}$$

- اگر اضلاع مثلث اعداد تک‌رقمی باشند، تعداد مثلث‌هایی که می‌توان تشکیل داد را نمایش می‌دهد. سپس اضلاع این مثلث‌ها را نمایش می‌دهد (زمانی اضلاع تشکیل مثلث را می‌دهند که مجموع هر دو ضلع بیش‌تر از ضلع سوم باشد).

- اگر اضلاع مثلث اعداد ۱ تا ۹ (اعداد تک‌رقمی) باشند، برنامه‌ای که تعداد مثلث متساوی‌الساقین که با این اعداد می‌توان تشکیل داد را نمایش می‌دهد.

- اگر اضلاع مثلث اعداد تک‌رقمی باشند، تعداد مثلث‌های قائم‌الزاویه که می‌توان تشکیل داد را نمایش می‌دهد. سپس اضلاع این مثلث را نمایش می‌دهد (زمانی که سه ضلع  $a$ ,  $b$  و  $c$  تشکیل مثلث قائم‌الزاویه را می‌دهند که  $a^2 + b^2 = c^2$  یا  $a^2 + c^2 = b^2$  یا  $b^2 + c^2 = a^2$  باشد).

- برنامه‌ای که مقدار سه زاویه را برحسب درجه دریافت کرده، تشخیص می‌دهد که آیا زاویه تشکیل مثلث را می‌دهند یا نه؟

➤ برنامه‌ای که حرفی را خوانده، با استفاده از Switch تشخیص می‌دهد حروف صدا دار است یا بی‌صدا؟

➤ برنامه‌ای که دو عدد را خوانده، با استفاده از Swlch عدد بزرگ‌تر را پیدا کرده و نمایش می‌دهد.

➤ برنامه‌ای که ضرایب یک معادله درجه ۲ را خوانده، با استفاده از سوئیچ ریشه‌های آن را محاسبه کرده و نمایش می‌دهد.

➤  $\Delta = b^2 - 4ac$   $ax^2 + bx + c = 0$

اگر  $\Delta > 0$  باشد،  $\frac{-b + \sqrt{\Delta}}{2a}$  و  $\frac{-b - \sqrt{\Delta}}{2a}$  دو ریشه هستند.

اگر  $\Delta = 0$  باشد، معادله دارای یک ریشه  $\frac{-b}{2a}$  است وگرنه، معادله ریشه ندارد یا معادله

دارای ریشه‌های  $\frac{-b}{2a} + \frac{i\sqrt{-\Delta}}{2a}$  و  $\frac{-b}{2a} - \frac{i\sqrt{-\Delta}}{2a}$  است.

➤ برنامه‌ای که عددی را خوانده، تشخیص می‌دهد آیا عددی Strong است یا خیر؟ عددی Strong است که حاصل جمع فاکتوریل ارقام آن برابر خود آن عدد باشد. به عنوان مثال، (عدد ۱۴۵،  $1! + 4! + 5! = 145$ ) یک عدد Strong است.

➤ برنامه‌ای که عددی را خوانده تشخیص می‌دهد عددی Armstrong است یا خیر؟ عددی Armstrong است که مجموع ارقام به توان تعداد ارقام برابر آن عدد باشد. عدد زیر Armstrong است:

$$371 = 3^3 + 7^3 + 1^3 = 371$$

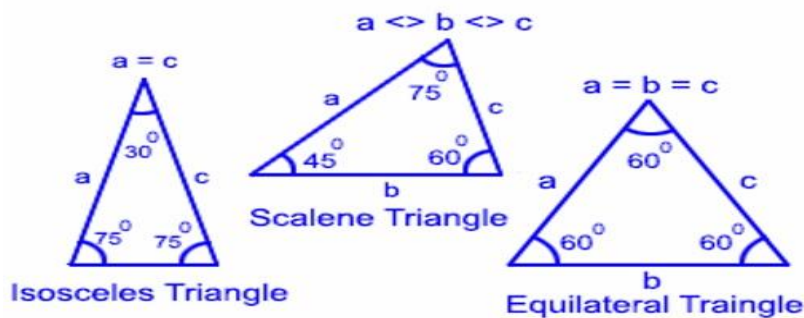
➤ برنامه‌ای که درجه حرارت را به سانتی گراد خوانده و بر اساس جدول زیر پیغام مناسب چاپ می‌کند.

پیغام	درجه حرارت
Freezing Weather	$< 0$
Very Cold Weather	$0 - 10$
Cold Weather	$10 - 20$
Normal	$20 - 30$
Hot	$30 - 40$
Very Hot	$> = 40$

- برنامه‌ای یک عدد در مبنای ۲ را خوانده به مبنای ۱۰ تبدیل می‌کند و نمایش می‌دهد. به‌عنوان مثال عدد زیر از مبنای ۲ به مبنای ۱۰ تبدیل شده است:

$$1100101 = 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 0 \times 2^4 + 1 \times 2^5 + 1 \times 2^6 = 1 + 0 + 4 + 0 + 0 + 32 + 64 = 10^3$$

- برنامه‌ای که اندازه زوایای یک مثلث را گرفته و نوع مثلث را تشخیص می‌دهد. اگر مجموع زوایای مثلث ۱۸۰ درجه نباشد، این زوایا تشکیل مثلث را نمی‌دهند. اما، اگر سه زاویه مثلث برابر باشند، مثلث متساوی‌الاضلاع است، ولی اگر اندازه دو زاویه مثلث برابر باشد، مثلث متساوی‌الساقین است، و گرنه مثلث مختلف‌الاضلاع است (مانند شکل زیر):



- برنامه‌ای که تمام اعداد سه‌رقمی که می‌توان با ارقام ۳ و ۴ و ۵ تولید کرد را نمایش می‌دهد (اعداد را به‌صورت صعودی مرتب‌شده نمایش می‌دهد).
- برنامه‌ای که تمام اعداد چهاررقمی که با ارقام ۵، ۶، ۷ و ۸ می‌توان تولید کرد را نمایش می‌دهد (در این برنامه در تولید اعداد از ارقام تکراری استفاده نمی‌کند).
- برنامه‌ای که اعداد سه‌رقمی با ارقام فرد می‌توان تولید کرد را نمایش می‌دهد که ارقام آن تکراری نیستند.
- برنامه‌ای که تمام اعداد سه‌رقمی را تولید می‌کند که رقم یکان آن اعداد زوج ۲ تا ۸، رقم دهگان آن اعداد فرد ۵ تا ۹ و رقم یکان آن اعداد مضرب ۳ تا ۹ می‌باشند را نمایش می‌دهد.
- برنامه‌ای که قیمت خرید و فروش یک کالا را خوانده، میزان سود یا زیان را محاسبه کرده، نمایش می‌دهد.
- یک اسب‌سوار به یک پیرزن برخورد کرده، تمام تخم مرغ‌های پیرزن از سبد می‌افتد و می‌شکند. صاحب اسب از پیرزن عذرخواهی می‌کند و به او می‌گوید خسارت تخم

مرغ‌هایتان را می‌دهم. پیرزن می‌گوید، تخم‌مرغ‌ها را دو تا دو تا برداشتم، یکی ماند، سه تا سه برداشتم، نیز یکی ماند، چهار تا چهار تا برداشتم، همچنین یکی ماند، پنج تا پنج تا برداشتم، یکی ماند، شش تا شش تا برداشتم، نیز یکی ماند، ولی هفت تا هفت تا برداشتم، هیچ تخم مرغی نماند. برنامه‌ای که تعیین می‌کند، حداقل چند تخم مرغ در سبد پیرزن وجود داشت.

➤ برنامه‌ای که تا زمانی که کاربر کلید ESC را فشار ندهد، کلید از کاربر می‌گیرد. به محض این که کاربر کلید ESC را فشار دهد، از برنامه خارج می‌شود. این برنامه کلید و کد اسکی کاراکتر فشار داده شده را نمایش می‌دهد. کد اسکی کلید ESC، ۲۷ است.

➤ یک رابطه ریاضی خیلی خوب برای تخمین جذر اعداد به این صورت است که ابتدا یک متغیر  $t$  را برابر ۱ قرار می‌دهیم، سپس هر بار  $t$  جدید را از روی  $t$  قدیم این گونه محاسبه می‌کنیم:

$$t_{\text{قدیم}} + \frac{x}{t_{\text{قدیم}}} * 0,5 = t_{\text{جدید}}$$

به این ترتیب اگر این کار را به صورت متوالی انجام دهیم، به تعمیم خوبی برای  $\sqrt{x}$  خواهیم رسید. برنامه‌ای که  $x$  و  $n$  را خواند و تا  $n$  مرحله تخمین  $\sqrt{x}$  را پیش می‌برد (تخمین نهایی  $\sqrt{x}$  همان جدید  $t$  آخرین مرحله خواهد بود).

➤ برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه‌های تکرار تودرتو و کاراکتر '\*' حرف 'A' را نمایش می‌دهد.

➤ برنامه‌ای که تعداد سطر (عددی زوج) را خوانده و با استفاده از حلقه تکرار تو در تو و کاراکتر '\*' حرف 'B' را نمایش می‌دهد.

➤ برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تو در تو و کاراکتر '\*' حرف 'C' را نمایش می‌دهد.

➤ برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تکرار تو در تو و کاراکتر '\*' کاراکتر 'D' را نمایش می‌دهد.

➤ برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تکرار تو در تو و کاراکتر '\*' کاراکتر 'E' را نمایش می‌دهد.

- برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تکرار تو در تو و '\*' کاراکتر 'F' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تکرار تو در تو و کاراکتر '\*' کاراکتر 'G' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تکرار تو در تو و کاراکتر '\*' کاراکتر 'H' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تکرار تو در تو و کاراکتر '\*' کاراکتر 'I' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '\*' کاراکتر 'J' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تکرار تو در تو و کاراکتر '\*' کاراکتر 'K' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده و با استفاده از حلقه تو در تو و کاراکتر '\*'، کاراکتر 'L' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، و با استفاده از حلقه تکرار تو در تو و کاراکتر '\*'، کاراکتر 'M' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '\*'، کاراکتر 'N' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '\*'، کاراکتر 'O' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '\*'، کاراکتر 'P' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '\*'، کاراکتر 'Q' را نمایش می‌دهد.

## ۶۰۰ برنامه C# با حل آن‌ها ۱۲۱

- برنامه‌ای که تعداد و سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '\*', کاراکتر 'R' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '\*', کاراکتر 'S' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '\*', کاراکتر 'T' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '\*', کاراکتر 'U' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '\*', کاراکتر 'W' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '\*', کاراکتر 'V' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از حلقه تکرار تو در تو و کاراکتر '\*', کاراکتر 'X' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطر را خوانده، با استفاده از کاراکتر '\*', کاراکتر 'Y' را نمایش می‌دهد.
- برنامه‌ای که تعداد سطرها را خوانده، با استفاده از کاراکتر '\*' کاراکتر 'Z' را نمایش می‌دهد.

**پیوست**

**الکترونیکی (فصل سوم)**

➤ برنامه‌ای که اعداد  $x$  و  $n$  را خوانده، حاصل سری زیر را محاسبه کرده و نمایش می‌دهد:

$$\frac{x^n 0^n}{-n} + \frac{x^1 (1)^{n-1}}{-(n-1)} + \dots + \frac{x^{(n-1)} (n-1)^1}{-1} + \frac{x^n n^1}{1} + \frac{x^{(n+1)} (n+1)^2}{2} + \dots + \frac{x^{(2n-1)} (2n-1)^n}{n}$$

➤ برنامه‌ای که  $n$  را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\frac{1}{1^2+1} + \frac{2}{2^2+2} + \frac{3}{3^2+3} + \dots + \frac{n}{n^2+n}$$

➤ برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\frac{\tan x}{\sin(x^n)} + \frac{\tan^2(x)}{\sin(x^{n-1})} + \frac{\tan^3 x}{\sin(x^{n-2})} + \dots + \frac{\tan^n x}{\sin(x)}$$

➤ برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل عبارت زیر را نمایش می‌دهد:

$$\overleftrightarrow{\log(\log(\log \dots n \dots \log(\log(x)))}$$

➤ برنامه‌ای که  $n$  را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\frac{1!}{1^2} + \frac{2!}{1^2+2^3} + \frac{3!}{1^2+2^3+3^4} + \dots + \frac{n!}{1^2+2^3+\dots+n^{n+1}}$$

➤ برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots \pm \frac{(-1)^n x^{2n}}{(2n)!}$$

➤ برنامه‌ای که  $x$  (بین -۱ تا ۱) و  $n$  را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\arcsin x = x + \frac{2!x^3}{4^1(1!)^2(3)} + \frac{4!x^5}{4^2(2!)^3(5)} + \dots + \frac{(2^n)!x^{2n+1}}{4^n(n!)^2(2n+1)}$$

➤ برنامه‌ای که  $n$  را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\frac{\sqrt{1}}{1} - \frac{1!}{1^2} + \frac{\sqrt{2}}{2} - \frac{2!}{2^2} + \frac{\sqrt{3}}{3} - \frac{3!}{3^2} + \frac{\sqrt{n}}{n} - \frac{n!}{n^2}$$

➤ برنامه‌ای که n را خوانده، حاصل عبارت زیر را نمایش می‌دهد:

$$\frac{\sqrt{3}-\sqrt{2}}{\sqrt{2^2+2}} + \frac{\sqrt{4}-\sqrt{3}}{\sqrt{3^2+3}} + \frac{\sqrt{n+1}-\sqrt{n}}{\sqrt{n^2+n}}$$

➤ برنامه‌ای که n را خوانده، حاصل عبارت زیر را نمایش می‌دهد:

$$\frac{1}{2^1(\log 2)^2} + \frac{1}{3^2(\log 3)^3} + \dots + \frac{1}{n^{n-1}(\log n)^n}$$

➤ برنامه‌ای که n را خوانده، حاصل عبارت زیر را نمایش می‌دهد:

$$\log \frac{2}{2^2-1} + \log \frac{3^2}{3^3-1} + \dots + \log \left( \frac{n^{n-1}}{n^n-1} \right)$$

➤ برنامه‌ای که n را خوانده، حاصل عبارت زیر را نمایش می‌دهد:

$$\frac{\sqrt{5}}{(2^3+2)^2} + \frac{\sqrt{7}}{(3^3+3)^2} + \dots + \frac{\sqrt{2n+1}}{(n^3+n)^2}$$

➤ برنامه‌ای که n را خوانده، حاصل عبارت زیر را محاسبه می‌کند و نمایش می‌دهد:

$$\frac{\sin(\frac{1}{2})}{\cos(\frac{1}{1}) \cdot \cos(\frac{1}{2})} + \frac{\sin(\frac{1}{6})}{\cos(\frac{1}{2}) \cdot \cos(\frac{1}{3})} + \dots + \frac{\sin\left(\frac{1}{n(n+1)}\right)}{\cos(\frac{1}{n}) \cdot \cos\left(\frac{1}{n+1}\right)}$$

➤ برنامه‌ای که x و n را خوانده، حاصل سری زیر را نمایش می‌دهد:

$$\frac{\sqrt{\sin(n) \cdot \log(2!)}}{\tan^n(1)} + \frac{\sqrt{\sin(n-1) \cdot \log(3!)}}{\tan^{n-1}(2)} + \dots + \frac{\sqrt{\sin(o) \cdot \log(n+2)!}}{\tan^n(n+1)}$$

➤ متدی که x و n را به عنوان پارامتر دریافت کرده، حاصل سری زیر را برمی‌گرداند.

این برنامه از این متد استفاده می‌کند.

$$\frac{\tan(x^1) + \tan(x^2) + \dots + \tan(x^n)}{\sin^1(x) + \sin^2(x) + \dots + \sin^n(x)}$$

➤ متدی n را به عنوان دریافت کرده، سپس حاصل سری زیر را محاسبه و نمایش می‌دهد.

برنامه‌ای که از این متد استفاده می‌کند.

$$(n-1)! \cdot 1^{(n-1)} + (n-2)! \cdot 2^{(n-2)} + \dots + 0! \cdot n^{(n-n)}$$

- متدی که عدد صحیح  $n$  را به عنوان پارامتر دریافت کرده، حاصل سری زیر را محاسبه می کند و برنامه اصلی آن را نمایش می دهد:

$$\frac{1}{n-1}x + \frac{n-2}{2}x + \dots + \frac{n-1}{1}x$$

- برنامه ای که  $n$  را خوانده با استفاده از متدی حاصل سری زیر را محاسبه و نمایش می دهد:

$$\frac{1*2*3}{(n+1)(n+2)(n+3)} + \frac{2*3*4}{(n+2)(n+3)(n+4)} + \dots + \frac{n(n+1)(n+2)}{(n+n)(n+n+1)(n+n+2)}$$

- برنامه ای که  $x$  و  $n$  را خوانده و توسط متدهای حاصل سری زیر را محاسبه کرده، نمایش می دهد:

$$1 + \frac{nx^1}{1!} + \frac{n(n-1)x^2}{2!} + \dots + \frac{n!x^n}{n!}$$

- برنامه ای که  $n$  را خوانده و با استفاده از متدی حاصل سری زیر را محاسبه کرده و نمایش می دهد:

$$\phi = \left\{ \begin{array}{l} \frac{1}{1 + \frac{1}{1+1}} \\ \dots \\ 1 + \frac{1}{1+1} \end{array} \right.$$

جمله  $n$

- برنامه ای که عدد  $n$  را خوانده، کلیه اعداد مضرب  $\gamma$  زیر  $n$  که یکی از ارقام آن  $\gamma$  باشد را نمایش می دهد. در این برنامه برای نمایش اعداد مضرب  $\gamma$  یک متد و برای تعیین این که یک رقم عدد  $\gamma$  است از متد دیگری استفاده می شود.

- برنامه ای که عدد  $n$  را خوانده، کلیه اعداد مضرب  $\gamma$  زیر  $n$  که هیچ یک از ارقام آن  $\gamma$  نباشد را نمایش می دهد.

## ۶۰۰ برنامه C# با حل آن ها ۱۲۵

➤ برنامه ای که عددی را خوانده، اگر ارقام از سمت راست به چپ دوتا دوتا کم شدند، Yes، وگرنه No را نمایش می دهد. برای تعیین این که اعداد از راست به چپ دو تا کم شدند از یک متد استفاده می کند. به عنوان مثال، برای اعداد ۲۴۶، ۳۵۷ و ۱۳۵، "Yes" و برای اعداد ۱۲۴، ۳۵۶، 'No' را نمایش می دهد.

- برنامه‌ای که رشته‌ای را خوانده، با استفاده از یک متد فضای خالی سمت چپ (ابتدا رشته) را حذف می‌کند و نمایش می‌دهد (کد اسکی کاراکترهای خالی ۳۲ است).
- برنامه‌ای که رشته‌ای را خوانده، با استفاده از اشاره‌گر و بازگشتی، طول رشته را حساب کرده، نمایش می‌دهد.
- برنامه‌ای که رشته‌ای را خوانده، با استفاده از اشاره‌گر و متد بازگشتی معکوس رشته را نمایش می‌دهد.
- برنامه‌ای دو نام را خوانده، تعیین می‌کند که آیا دو نام با هم برابر هستند یا خیر؟
- برنامه‌ای که رشته‌ای را از ورودی خوانده و بین حروف آن فاصله (Blank) ایجاد می‌کند. به عنوان مثال، اگر کاربر رشته Test را وارد کند، برنامه رشته T e s t را نمایش می‌دهد. در این برنامه متغیرهای  $S_1$  (رشته ورودی) و  $S_2$  (رشته خروجی) را داریم.
- برنامه‌ای که ابتدا  $n$  را خوانده، سپس  $n$  رشته را می‌خواند و طول هر یک را نمایش می‌دهد.
- برنامه‌ای که ابتدا  $n$  را خوانده، سپس  $n$  رشته را می‌خواند و رشته‌ای که بیشترین طول را دارد، نمایش می‌دهد.
- برنامه‌ای که ابتدا  $n$  را خوانده، سپس  $n$  رشته را می‌خواند و رشته‌هایی که حرف اول و آخر آن‌ها یکسان باشد را نمایش می‌دهد.
- برنامه‌ای که ابتدا  $n$  را خوانده، سپس  $n$  رشته را خوانده و در پایان یک کاراکتر را می‌خواند و رشته‌هایی که با این کاراکتر شروع می‌شوند یا پایان می‌یابند را نمایش می‌دهد.
- برنامه‌ای که یک رشته را خوانده، سپس یک کاراکتر را می‌خواند و بین حروف رشته آن کاراکتر را اضافه می‌کند.

## ۶۰۰ برنامه C# با حل آن‌ها ۱۲۷

- برنامه‌ای که رشته‌های  $S_1$  و  $S_2$  را خوانده و کاراکترها را یک‌درمیان در رشته  $S_3$  قرار می‌دهد و رشته  $S_3$  را نمایش می‌دهد.
- برنامه‌ای که رشته‌ای را خوانده، کاراکتری از رشته که بیش‌ترین کد اسکی را دارد، نمایش می‌دهد.
- برنامه‌ای که رشته‌ای را از ورودی خوانده و تمام کاراکترهای تکراری آن را حذف می‌نماید.
- برنامه‌ای که دو رشته را خوانده، حروف مشترک بین آن‌ها را نمایش می‌دهد.
- برنامه‌ای که رشته‌ای را خوانده، سپس دو عدد را می‌خواند و با توجه به اعداد خوانده‌شده، حروف آن مکان‌ها را جابجا نماید. در این برنامه از متد `Change` به‌صورت زیر استفاده می‌شود:
  - `String Change (String s, int n1, int n2),`  
به‌عنوان مثال، اگر متد `Change` به‌صورت زیر فراخوانی شود:  
`Change ("Fanavarienovin", ۱, ۳)`  
متد، رشته `"naFavarienovin"` را برمی‌گرداند.
- برنامه‌ای که تعداد فاصله اضافی به سمت چپ رشته اضافه می‌کند تا اندازه رشته به  $n$  تغییر یابد. اگر اندازه رشته بزرگ‌تر یا مساوی  $n$  باشد، رشته هیچ تغییر نمی‌یابد.
- برنامه‌ای که حروف 'a' تا 'z' و 'A' تا 'Z' را به همراه کد اسکی آن‌ها نمایش می‌دهد.
- برنامه‌ای که رشته‌ای را خوانده و کاراکتری که بیش‌ترین تکرار را دارد، نمایش می‌دهد.
- برنامه‌ای که دو رشته را خوانده، رشته دوم را در وسط رشته اول اضافه می‌کند.
- برنامه‌ای که رشته‌ای را خوانده و بعد از هر یک از حروف کوچک، حروف بزرگ آن کاراکتر را اضافه می‌کند. به‌عنوان مثال، رشته `"Tree"` به رشته `"TrReEeE"` تبدیل می‌شود.
- برنامه‌ای که دو رشته را دریافت کرده و رشته دوم را بعد از  $n$  امین کاراکتر رشته اول درج می‌نماید. در این برنامه متد `insert` استفاده‌شده که دارای اعضای زیر است:
  - `string insert (string s1, string s2, int n);`

➤ برنامه‌ای که تعدادی رشته را خوانده (برای خاتمه برنامه رشته پایانی Finish وارد می‌گردد)، تعداد رشته‌هایی که در آن‌ها کاراکترهای رقمی ('۰' تا '۹') وجود دارد را شمارش می‌کند.

➤ برنامه‌ای که یک ماتریس را خوانده، تشخیص می‌دهد که آیا این ماتریس یک ماتریسی اسپارس است یا نه؟ ماتریسی اسپارس است که حداقل نصف عناصر آن صفر باشند. یعنی، اگر ماتریس  $n \times m$  باشد، حداقل  $(n*m)/2$  عناصر آن صفر باشد.

➤ برنامه‌ای که یک ماتریس را می‌خواند و تعیین می‌کند که آیا متقارن است یا خیر؟ ماتریسی متقارن است که برابر ترانواده خودش باشد. یعنی  $A=A^T$  به عنوان مثال، ماتریس زیر متقارن است:

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 8 \end{bmatrix}^T$$

➤ برنامه‌ای که یک عدد را خوانده، تعداد هر رقم استفاده شده در آن را نمایش می‌دهد.

➤ برنامه‌ای که یک عدد مبنای ۲ را به صورت رشته خوانده، متمم یک آن را نمایش می‌دهد.

➤ برنامه‌ای که یک عدد باینری را دریافت کرده، به مبنای ۸ تبدیل می‌کند و نمایش می‌دهد. برای تبدیل عدد از باینری به مبنای ۸ عدد را سه رقم سه رقم جدا کرده و معادل مبنای ۸ را جایگزین هر سه رقم می‌کند (مطابق جدول زیر):

معادل مبنای ۸	معادل باینری	معادل مبنای ۸	معادل مبنای ۲
۰	۰۰۰	۱	۰۰۱
۲	۰۱۰	۳	۰۱۱
۴	۱۰۰	۵	۱۰۱
۶	۱۱۰	۷	۱۱۱

اگر عدد باینری ۰۱۱۰۱۱۰۰ باشد، معادل آن ۶(۱۰۱)، ۵(۱۰۱) و ۱(۰۱) است (یعنی معادل ۱۵۶)

➤ برنامه‌ای که عددی در مبنای ۲ خوانده، به مبنای ۱۶ تبدیل می‌نماید و نمایش می‌دهد. هر چهار بیت مبنای ۲ معادل یک رقم مبنای ۱۶ است (معادل جدول زیر):

مبنا	مبنا	مبنا	مبنا	مبنا	مبنا	مبنا	مبنا

## ۶۰۰ برنامه C# با حل آن‌ها ۱۲۹

ی	۱۶	ی	۱۶	ی	۱۶	ی	۱۵	ی	۱۶
۰	۰	۰	۱	۰	۰	۰	۳	۰	۰
۴	۰	۰	۵	۰	۰	۰	۷	۰	۰
۸	۰	۰	۹	۰	۰	۰	B	۰	۰
C	۰	۰	D	۰	۰	۰	F	۰	۰

به عنوان مثال، عدد باینری ۱۰۱۱۱۱۱۰۱۰۱۰۱ معادل ۱۷۰۵ (۱=۰۰۰۱, ۷=۰۱۱۱, E=۱۱۰۱, ۵=۰۱۰۱) است.

➤ برنامه‌ای که یک آرایه با  $n$  عنصر و سه عدد  $x, y, z$  را خواند و مقدار ماکزیمم  $(x*a[i])+(y*a[j])+(z*a[k])$  را برای  $i \leq j \leq k$  پیدا کرده و نمایش می‌دهد.

➤ برنامه‌ای که رشته‌ای را خوانده، کلمات آن را از انتها به ابتدای رشته نمایش می‌دهد. به عنوان مثال، اگر ورودی به صورت زیر باشد:

- I Like Cat Programming

خروجی به صورت زیر خواهد بود.

- Programming Cat Like I

➤ برنامه‌ای که رشته‌ای را خوانده (TXT) و سپس یک الگو (Pat) را می‌خواند و در رشته جست‌وجو می‌نماید. مکان‌های وقوع Pat در TXT را نمایش می‌دهد.

به عنوان مثال، اگر

`TXT= "AABAACAADAABAABA"`

`Pat= "AAB"`

باشد، خروجی به صورت زیر نمایش داده شود:

Pattern Found at index ۰

Pattern Found at index ۹

Pattern Found at index ۱۲

➤ برنامه‌ای که آرایه‌ای با  $n$  عنصر که مقادیر آن‌ها ۰ یا یک به ترتیب تصادفی هستند را خوانده، ۰های را به سمت چپ و ۱ها را به سمت راست آرایه منتقل کرده، نمایش می‌دهد. به عنوان مثال، اگر آرایه ورودی به صورت زیر باشد:

[۰, ۱, ۰, ۰, ۱, ۱, ۰, ۱, ۰, ۰]

خروجی به صورت زیر خواهد بود:

[۱, ۱, ۱, ۱, ۱, ۱, ۱, ۱, ۱, ۱]

➤ برنامه‌ای که آرایه‌ای را خوانده، سپس سه عدد  $x, y$  و  $z$  را می‌خواند و ماکزیمم عبارت زیر را نمایش می‌دهد:

$$\max_{i \leq j \leq k} ((x * a[i]) + (y * a[j]) + (z * a[k]))$$

به‌عنوان مثال، اگر آرایه  $a$  به‌صورت زیر باشد:

$$a[] = \{-1, -2, -3, -4, -5\}$$

$x=1$ ،  $y=2$  و  $z=-3$  باشد، آنگاه خروجی ۱۲ خواهد بود. زیرا:

$$(1 * -1) + (2 * -2) + (-3 * -5) = 12$$

➤ برنامه‌ای که یک آرایه را خوانده، دو عنصری را می‌یابد که حاصل جمع آن‌ها نزدیک‌ترین عدد به صفر است. برای انجام این برنامه باید عددی را پیدا کنیم که مینیمم قدر مطلق مجموع دو عنصر آرایه می‌باشد.

➤ برنامه‌ای که آرایه‌ای را خوانده، ماکزیمم مقدار اختلاف بین هر دو عنصر آرایه را نمایش می‌دهد.

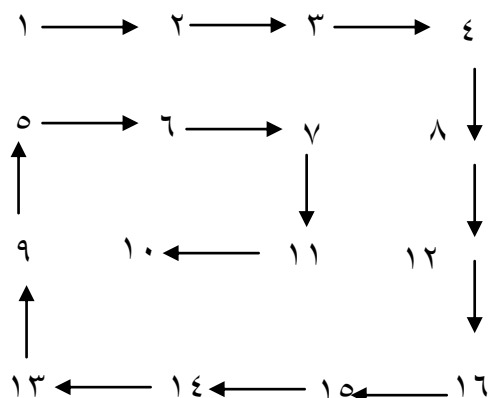
➤ برنامه‌ای که آرایه‌ای را خوانده، اعداد فرد و زوج را از هم جدا می‌کند.

➤ برنامه‌ای که آرایه‌ای را خوانده و ماکزیمم  $i-j$  را پیدا می‌کند به‌طوری‌که  $arr[j] > arr[i]$  باشد. سپس مقدار ماکزیمم را نمایش می‌دهد. به‌عنوان مثال اگر آرایه به‌صورت زیر باشد:

$$arr[] = \{۳۴, ۸, ۱۰, ۳, ۲, ۸۰, ۳۰, ۳۳, ۱\}$$

آنگاه مقدار ۶ را نمایش می‌دهد وقتی که  $(j=۷, i=۱)$  است.

➤ برنامه‌ای که یک آرایه دوبعدی تعریف کرده و مقداردهی اولیه می‌کند. سپس اطلاعات آرایه را به‌صورت مارپیچی نمایش می‌دهد. به‌عنوان مثال، اگر آرایه دوبعدی به‌صورت زیر باشد:



خروجی به صورت زیر است:

➤ ۱ ۲ ۳ ۴ ۸ ۱۲ ۱۶ ۱۵ ۱۴ ۱۳ ۹ ۵ ۶ ۷ ۱۱ ۱۰

➤ برنامه‌ای که یک آرایه دوبعدی را تعریف کرده، مقداردهی می‌کند (تمام عناصر آرایه ۰ (False z) یا ۱ (True) هستند)، اگر عنصر  $a[i][j]$  برابر یک (۱) باشد، تمام عنصر سطر  $i$  ام و ستون  $j$  ام را برابر یک قرار می‌دهد. به عنوان مثال، اگر آرایه به صورت زیر باشد:

```
0 1 0
0 0 0
0 0 1
```

خروجی به صورت زیر است:

```
1 1 1
0 1 1
0 1 1
```

➤ برنامه‌ای که مقادیر چند آرایه را مقداردهی اولیه می‌کند و یک نقطه ثابت شده (عنصری از آرایه که مقدار آن عنصر برابر اندیس آن است) را نمایش می‌دهد. چنانچه نقطه ثابت نباشد، ۱- را نمایش خواهد داد. به عنوان مثال، آرایه‌های زیر را در نظر بگیرید:

```
arr۱ = {-۱۰, -۵, ۰, ۳, ۷}
arr۲ = {۱۰, ۲, ۵, ۸, ۱۷}
```

برای  $arr_1$  خروجی ۳ (چون  $arr_1[3]=3$  است) و برای  $arr_2$  خروجی ۱- است.

➤ برنامه‌ای که آرایه‌ای را تعریف کرده و خوانده، به طوری که مقدار عناصر آن ابتدا افزایشی است (یعنی، عنصر قبلی کم‌تر از عنصر بعدی است) و سپس عناصر آن کاهش‌ی است. این برنامه بزرگ‌ترین مقدار این نوع آرایه را نمایش می‌دهد. به عنوان مثال، اگر عناصر آرایه به صورت زیر باشد:

۷ ۴۱ ۱ ۳ ۵۰ ۱۰

خروجی ۵۰ خواهد بود.

➤ برنامه‌ای که یک آرایه را تعریف کرده، سپس می‌خواند و در پایان، آرایه‌ای را ایجاد می‌کند که هر عنصر آن، تعداد عناصری کوچک‌تر از سمت راست خودش را شمارش می‌کند. به عنوان مثال، اگر آرایه به صورت زیر باشد:

arr={۱۲, ۱, ۲, ۳, ۰, ۱۱, ۴}

خروجی به صورت زیر خواهد بود:

۶, ۱, ۱, ۱, ۰, ۱, ۰

➤ برنامه‌ای که آرایه‌ای را تعریف کرده، و می‌خواند. سپس یک عدد را خوانده و آرایه فرعی را نمایش می‌دهد که مجموع عناصر آن برابر عدد وارد شده می‌باشد. اگر آرایه‌ای وجود نداشته باشد که مجموع عناصر آن برابر عدد وارد شده نباشد، پیام "No SubArray Found" را نمایش می‌دهد. به عنوان مثال، اگر آرایه‌های ورودی به صورت زیر باشد:

۵ ۱ ۴ ۲۰ ۰ ۳ ۱۰

و اگر sum=۳۰ باشد، آنگاه خروجی به صورت زیر است:

➤ Sum found between indexes ۱ and ۴

➤ برنامه‌ای که یک آرایه را تعریف کرده و می‌خواند، سپس یک عدد را می‌خواند، توسط متدی اگر مجموع سه عنصر آرایه برابر با عدد خوانده شده باشد، آن سه عدد را نمایش می‌دهد و True را برمی‌گرداند.

➤ برنامه‌ای که کوچک‌ترین عدد مثبت (غیر منفی) که در آرایه غیر مرتب حذف شده ست را می‌یابد و نمایش می‌دهد. به عنوان مثال، اگر آرایه ورودی به صورت زیر باشد:

{۲, ۳, -۷, ۶, ۸, ۱, -۱۰, ۱۵}

خروجی عدد ۴ خواهد بود. ولی برای آرایه زیر خروجی ۲ خواهد شد:

{۱, ۱, ۰, -۱, -۲}

➤ برنامه‌ای که آرایه‌ای را تعریف کرده، خوانده، سپس یک عدد را می‌خواند و زوج مقادیر از عناصر آرایه که اختلاف آن‌ها عدد وارد شده باشد، را نمایش می‌دهد. به عنوان مثال، اگر عناصر آرایه به صورت زیر باشد:

{۲, ۲۰, ۳, ۲, ۵۰, ۸۰}

و عدد ورودی ۷۸ باشد، خروجی به صورت زیر است:

Pair Found: (۲, ۸۰)

➤ برنامه‌ای که آرایه‌ای را تعریف کرده، می‌خواند، سپس هر عنصر را با بزرگ‌ترین عنصر بعد از خودش (سمت راست خودش) جایگزین می‌کند، به جای آخرین عدد ۱- قرار می‌گیرد و آرایه جدید را نمایش می‌دهد. به عنوان مثال، اگر آرایه ورودی به صورت زیر باشد:

### ۶۰۰ برنامه C# با حل آن‌ها ۱۳۳

{ ۱۶, ۱۷, ۴, ۳, ۵, ۲ }

آرایه خروجی به صورت زیر خواهد شد:

{ ۱۷, ۵, ۵, ۵, ۲, - }

➤ برنامه‌ای که آرایه‌ای را تعریف کرده، می‌خواند، سپس یک عدد را خوانده و مجموع ۴ عنصر آرایه که برابر با عدد خوانده شده باشد را نمایش می‌دهد. به عنوان مثال، اگر آرایه ورودی به صورت زیر باشد:

{ ۱۰, ۲, ۳, ۴, ۵, ۹, ۷, ۸ }

و عدد ورودی برابر ۲۳ باشد، آنگاه خروجی به صورت زیر است:

➤ ۳ ۵ ۷ ۸

➤ برنامه‌ای که آرایه‌ای را تعریف کرده و عناصر آن را مقداردهی اولیه می‌کند. سپس عناصری از آرایه که مقادیر آن‌ها تشکیل مثلث را می‌دهد را شمارش می‌کند (سه مقدار از آرایه که مجموع هر دو مقدار بیش‌تر از مقدار سوم باشد، تشکیل مثلث را می‌دهند).

➤ برنامه‌ای که کلاس کسر (Fraction) را پیاده سازی کرده، عملگرهای +، -، \*، /، =، !=، >، <، >= و <= را برای آن پیاده سازی می‌کند (کسر دارای صورت (x) و مخرج (y) است).

➤ برنامه‌ای که ایجاد و استفاده از متدهای توسعه یافته را نشان می‌دهد.

➤ برنامه‌ایی که دایرکتوری فعلی را نمایش می‌دهد (هدف این برنامه آشنایی با متدهای خارجی است).

➤ برنامه‌ای که اطلاعات کارمندان از قبیل نام، نام خانوادگی، نوع بیمه، مبلغ حقوق ماهانه و تعداد روزهایی که کار کرده را دریافت می‌کند، حقوق دریافتی کارمند را محاسبه کرده، برمی‌گرداند. حقوق دریافتی کارمند به صورت زیر محاسبه می‌شود:

حقوق دریافتی = بیمه - مالیات - تعداد روز \* حقوق کارمند

اگر نوع بیمه ۱ باشد، ۷ درصد \* حقوق ماهانه

اگر نوع بیمه ۲ باشد، ۹ درصد \* حقوق

بیمه ماهانه

اگر نوع بیمه ۳ باشد، ۱۰ درصد \* تعداد روز

اگر حقوق \* ماهانه بزرگتر از ۴۸۵۰۰۰ تومان باشد، ۱۰ درصد مازاد بر ۴۸۵۰۰۰ تومان مالیات منظور می‌شود.

توضیح: این برنامه با دو کلاس Person و Employee طراحی شده است. کلاس Person اطلاعاتی از قبیل نام، نام خانوادگی و نوع بیمه را نگهداری می‌کند و کلاس Employee از کلاس Person مشتق شده، علاوه بر اطلاعات نام، نام خانوادگی، و نوع بیمه که از کلاس Person به ارث می‌برد، اعضای داده‌ای salary و days را نیز دارد. در ادامه اعضای این کلاس‌ها را خواهیم دید.

➤ برنامه‌ای که کلاس نقطه (Point) را با استفاده از واسط پیاده‌سازی می‌کند (کلاس Point مختصات نقطه را به صورت دو عدد صحیح دریافت کرده، آن را نمایش می‌دهد).

➤ برنامه‌ای که با استفاده از کلاس static اعمال زیر را انجام می‌دهد:

○ فیلدی به نام number از نوع static تعریف می‌کند.

○ متدی به createRandom() تعریف کرده که دو عدد را به عنوان پارامتر دریافت می‌کند، یک عدد تصادفی بین دو عدد تولید می‌نماید.

## ۶۰۰ برنامه C# با حل آن‌ها ۱۳۵

- متدی به نام `isOdd()` تعریف می‌کند که تعیین می‌کند فیلد داده‌ای `number` فرد است یا خیر؟
- متدی به نام `isEven()` تعریف می‌کند که تعیین می‌کند فیلد داده‌ای `number` زوج است یا خیر؟
- متدی به نام `isSquare()` تعریف می‌کند که تعیین می‌کند فیلد داده‌ای `number` مربع کامل است یا خیر؟
- متدی به نام `isPerfect()` تعریف می‌کند که تعیین می‌کند فیلد `number` عددی کامل (تام) است یا خیر؟
- متدی به نام `isPrime()` تعریف می‌کند که تعیین می‌کند فیلد `number` عددی اول است یا خیر؟
- متدی به نام `Fact()` تعریف می‌کند که فاکتوریل فیلد `number` را برمی‌گرداند.

➤ برنامه‌ای که عملگرهای `+`، `-` و `*` را برای ماتریس‌های دو بعدی تعریف می‌کند.

➤ برنامه‌ای که مجموع مساحت و محیط چند شکل متفاوت از قبیل مربع، دایره، مستطیل و مثلث را محاسبه می‌کند. این برنامه کاربردهای مختلفی از قبیل محاسبه مساحت و محیط زمین برای ساخت و ساز یا فروش، محاسبه مساحت و محیط ساختمان جهت نقاشی یا لوله کشی دارد.

➤ امروزه در دانشگاه‌ها دو دسته دانشجو تحصیل می‌کنند: دانشجویان انتفاعی (روزانه) و دانشجویان غیر انتفاعی (مانند شبانه). دانشجویان روزانه هیچ مبلغی را پرداخت نمی‌کنند. اما، دانشجویان شبانه سه نوع مبلغ پرداخت می‌کنند که عبارتند از:

۱. مبلغ شهریه ثابت
۲. مبلغ شهریه متغیر برای هر واحد نظری
۳. مبلغ شهریه متغیر برای هر واحد عملی

برنامه‌ای که ابتدا تعداد دانشجو را از ورودی خوانده، سپس به اندازه تعداد دانشجو شماره دانشجویی، نام دانشجو، نوع دانشجو، تعداد واحدهای نظری و عملی دانشجو را دریافت کرده، مبلغ شهریه کل دانشجویان را نمایش می‌دهد (اگر نوع دانشجو صفر وارد شود، دانشجو روزانه وگرنه شبانه است).

➤ برنامه‌ای که کلاس مربع را از نوع کلاس `static` پیاده سازی می‌کند. این کلاس دارای عضو داده‌ای `side` (ضلع مربع)، عضو متدی `Area()` برای محاسبه مساحت مربع، عضو متدی `Perimeter()` برای محاسبه محیط مربع است (همان طور که در لیست برنامه می‌بینید، چون کلاس از نوع `static` است، همه این اعضا نیز `static` می‌باشند).

➤ برنامه‌ای که مجموع مساحت و محیط چند شکل متفاوت از قبیله مربع، دایره، مستطیل و مثلث را از طریق واسطه محاسبه می‌کند. این برنامه کاربردهای مختلفی از قبیل محاسبه مساحت و محیط زمین برای ساخت و ساز یا فروش، محاسبه مساحت و محیط ساختمان جهت نقاشی یا لوله کشی دارد.

## منابع

۱. جعفر نژاد قمی، عین اله، عباس نژادورزی، رمضان، برنامه نویسی به زبان C# (ویراست سوم)، بابل انتشارات علوم رایانه، ۱۳۸۸.
۲. عباس نژادورزی، رمضان، حل مسائل C++ (مرجع کامل)، بابل فن آوری نوین، ۱۳۸۸.
۳. عباس نژادورزی، رمضان، آموزش گام به گام برنامه نویسی بانک اطلاعاتی با C#، بابل فن آوری نوین، ۱۳۸۸.
۴. عباس نژادورزی، رمضان، رحیم پور کامی، باقر، هاشمیان، ابراهیم، طراحی سیستم‌های شی گرا به زبان C#، بابل فن آوری نوین، ۱۳۹۱.
۵. عباس نژادورزی، رمضان، یونسی، عمران، عباس نژادورزی، یوسف، حل C++ (آزمایشگاه کامپیوتر مرجع کامل)، بابل فن آوری نوین، ۱۳۹۶.
۶. Andrew Troelsen, "Pro C# ۲۰۰۸ and the .NET ۳,۵ Platform, Fourth Edition", Apress, ۲۰۰۷
۷. Daniel Solis, "Illustrated C# ۲۰۰۸", Apress, ۲۰۰۸
۸. JON SKEET, "C# in Depth", Manning, ۲۰۰۸
۹. Jesse Liberty & Alex Horovitz "Programming .NET ۳,۵", O'Reilly, ۲۰۰۸
۱۰. by Joe Duffy, "Professional .NET Framework ۲,۰", Wrox Press, ۲۰۰۶
۱۱. Donis Marshall, "Microsoft Visual C# ۲۰۰۸: The Language", Microsoft Press, ۲۰۰۸
۱۲. Richard Blum, "C# Network Programming", Sybex, ۲۰۰۳
۱۳. <http://barnamenevis.org>
۱۴. <http://www.codeproject.com/>
۱۵. <http://www.srco.ir/>

