

## پودمان دوم

# مدیریت مجموعه داده

هر مسئله دارای بخش‌های مختلفی، شامل داده، عملیات، حالت‌های شرط و تکرار است. تشخیص هر کدام از این بخش‌ها به حل درست مسئله منجر می‌شود. یک زبان برنامه‌نویسی برای پیاده‌سازی مسئله برای هر یک از این موارد، دستوراتی دارد که باید از خوانایی و سهولت استفاده برخوردار باشند. در برخی از مسئله‌ها نیاز به ایجاد ساختار تکرار و یا تعدادی داده هم‌منوع داریم که نمایش آنها در ساختار آرایه مناسب است. آرایه‌ها شامل داده‌هایی هم‌منوع و هم‌نام هستند که در برخی مسائل نظیر جست‌وجو و مرتب‌سازی استفاده از آنها ضروری است. زبان برنامه‌نویسی C# دارای روش‌های متنوعی برای کار با انواع دستورات شرط، تکرار و آرایه است. در این پودمان معرفی و کاربرد انواع ساختار تکرار، آرایه و فنون جست‌وجوی داده‌ها را فرا خواهید گرفت.

## واحد یادگیری ۳

### ■ شایستگی کار با ساختار تکرار

#### آیا تا به حال پی برده اید

- چرا در برخی برنامه‌ها، دستورات تکرار می‌شوند؟
  - چگونه می‌توان در برنامه، اجرای چند دستور را به تعداد معین تکرار کرد؟
  - چگونه می‌توان تعیین کرد تا زمان برقراری یک شرط اجرای چندین دستور تکرار شود؟
  - ساختار تکرار چگونه به بهینه‌سازی مسئله کمک می‌کند؟
- هدف از این واحد شایستگی به‌کارگیری ساختار تکرار در برنامه‌ها است.

#### استاندارد عملکرد

با استفاده از دانش ساختار تکرار در برنامه‌نویسی، مسئله را تحلیل و در صورت نیاز برنامه را با استفاده از ساختار تکرار کدنویسی و خطایابی کند.

## حلقه‌ها

آیا معلم از شما خواسته است تا جمع نمرات یک کلاس ۱۵ نفره را محاسبه کنید؟  
 آیا در زنگ ورزش ۱۰ بار دور حیاط مدرسه دویده‌اید؟  
 آیا در یک مسابقه فوتبال به عنوان یک گروه، فرصت زدن ۵ پنالتی به گروه مقابل را داشته‌اید؟  
 همه اینها شما را به یاد کارهای تکراری می‌اندازد. هرگاه یک یا چند عمل تکراری داشته باشیم، می‌توانیم از ساختاری به نام حلقه استفاده کنیم. این ساختار به شما کمک می‌کند تا یک یا چند دستور را یک‌بار بنویسید؛ ولی چندین بار اجرا کنید.  
 هنگامی که تعداد تکرار دستورات مشخص باشد حلقه معین است و در مواردی که از ابتدا تعداد تکرار دستورات مشخص نیست حلقه نامعین است.

به مثال‌های زیر توجه کرده و با کمک هنرآموز خود، مشخص کنید کدام یک از آنها معین یا نامعین هستند.

فعالیت گروهی



نوع حلقه (معین یا نامعین)	مثال
	یک نجار روزانه ۱۲ صندلی می‌سازد.
	تا زمانی که دبیر ورزش سوت پایان را بزند همه هنرجویان چندین نرمش کششی انجام می‌دهند.
	در یک نانوایی بربری، نانوا در هر تنور ۶۰ عدد نان می‌پزد.
	یک نقاش ساختمان در یک روز تعدادی دیوار با ابعاد مختلف را رنگ می‌زند.

برای آشنایی بیشتر با حلقه به مثال زیر توجه کنید:

مثال: یک هنرآموز، فعالیت منزل هنرجویان را تحویل گرفته، پس از بررسی گزارش هر کدام را در دفتر ارزشیابی خود وارد می‌کند. تعداد هنرجویان ۲۲ نفر است.

- دریافت فعالیت یک هنرجو
- مشاهده و بررسی فعالیت
- ثبت گزارش فعالیت، در دفتر ارزشیابی
- تحویل فعالیت بررسی‌شده به هنرجو
- تکرار مراحل بالا

	ورودی‌ها	شناخت مسئله
	خروجی‌ها	
دریافت فعالیت، بررسی فعالیت، ثبت گزارش برای هنرجو، تحویل فعالیت به هنرجو	بدنه حلقه	

شروع

$i \leftarrow 1$

$i \leq 22$

نادرست

درست

دریافت فعالیت یک هنرجو

ثبت گزارش فعالیت در دفتر ارزشیابی

پایان

۰- شروع

۱- در  $i$  مقدار یک قرار بده

۲- اگر  $i$  کوچک‌تر یا مساوی ۲۲ است برو به مرحله بعد، وگرنه برو به مرحله ۸

۳- فعالیت هنرجوی  $i$ ام را دریافت کن

۴- فعالیت را مشاهده و بررسی کن

۵- ثبت گزارش فعالیت، در دفتر ارزشیابی

۶- فعالیت به هنرجوی  $i$ ام تحویل داده شود

۷- به  $i$  یک واحد اضافه کرده و به مرحله ۲ برو

۸- پایان

برای شمارش هنرجویان از متغیر  $i$  استفاده می‌کنیم.

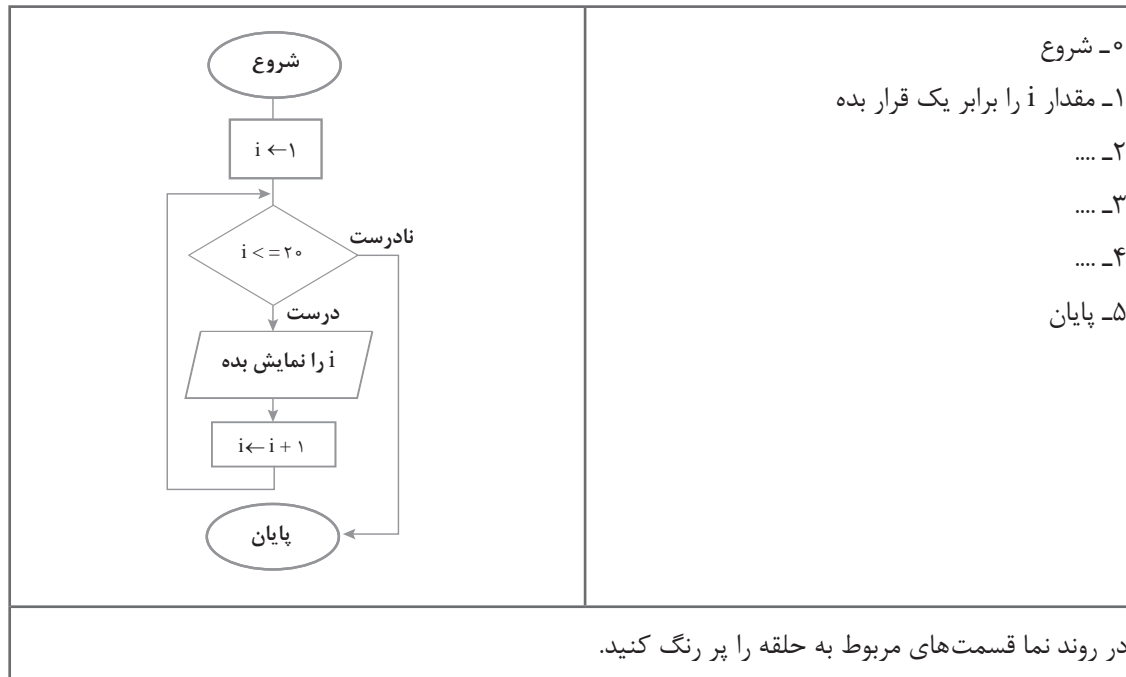
در روند نما قسمت‌های مربوط به حلقه را پر رنگ کنید.

روندنمای بالا را با توجه به الگوریتم برنامه تکمیل کنید.

فعالیت کارگاهی



مسئله: می‌خواهیم سیستم نوبت‌دهی یک مطب دندانپزشکی را مکانیزه کنیم، به‌طوری‌که برای نوبت‌دهی به ۲۰ بیمار خود از شماره‌های چاپ‌شده ۱ تا ۲۰ استفاده کند. با توجه به روندنما الگوریتم زیر را کامل کنید.



– اگر تعداد بیماران ۳۰ نفر باشد، چه تغییری لازم است؟  
 – روندنما و الگوریتم را تغییر دهید تا تعداد بیماران را از ورودی دریافت کند.

فعالیت کارگاهی



مسئله: شرط قبولی در یک آزمون کسب حداقل نمره ۱۲ از ۲۰ است. می‌خواهیم نمرات هنرجویان یک کلاس ۲۰ نفره را دریافت کرده، تعداد قبول شده‌ها را مشخص کنیم

ورودی‌ها	نمرات هنرجویان
شناخت مسئله	خروجی‌ها
	تعداد قبول شده‌ها
	بدنه حلقه
<p>۰- شروع</p> <p>۱- در شمارنده (COUNT) مقدار صفر قرار بده</p> <p>۲- در <math>i</math> مقدار یک قرار بده</p> <p>۳- .....</p> <p>۴- .....</p> <p>۵- اگر نمره بزرگ‌تر یا مساوی ۱۲ بود به شمارنده یکی اضافه کن</p> <p>۶- .....</p> <p>۷- برو به مرحله ۳</p> <p>۸- شمارنده را نمایش بده</p> <p>۹- پایان</p>	
در روند نما قسمت‌های مربوط به حلقه را پر رنگ کنید.	

## جدول ارزشیابی شایستگی‌های غیر فنی، ایمنی، بهداشت و توجهات زیست محیطی

شایستگی‌ها	شرایط عملکرد (ابزار، مواد، تجهیزات، زمان، مکان و ...)	نتایج ممکن	استاندارد (شاخص‌ها/داوری/نمره دهی)	نمره
شایستگی‌های غیر فنی	حل مسئله، شناسایی مسئله، تولید/ارزیابی راه‌حل‌ها - زبان فنی	قابل قبول	تعیین داده و اطلاعات مسئله - تعیین عملیات تکرار شونده در راه حل - ارائه راه‌حل خلاقانه برای مسئله - بررسی قابل اجرا بودن راه‌حل	۲
ایمنی و بهداشت	رعایت ارگونومی			
توجهات زیست محیطی	حفاظت از تجهیزات کارگاه			
نگرش	دقت در تعیین بدنه حلقه موردنیاز برای برنامه - استفاده از ابزارهای خطیابی برای رفع خطای برنامه	غیرقابل قبول	توجه به ایمنی و بهداشت محیط کارگاه	۱
● این شایستگی‌ها در ارزشیابی پایانی واحد یادگیری باید مورد توجه قرار گیرند.				

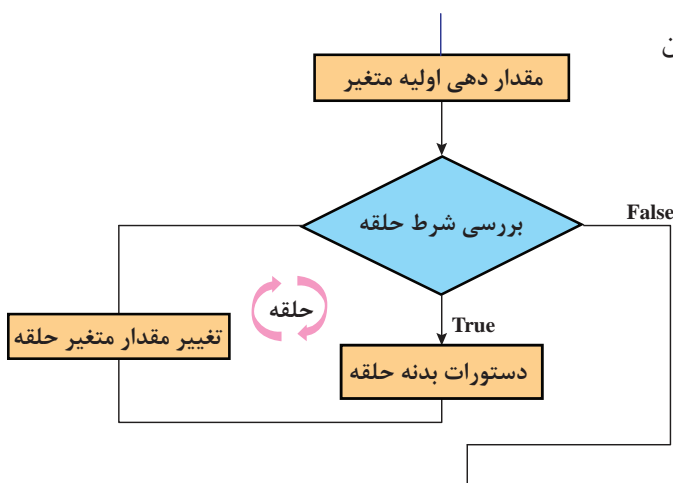




مراحل کار	شرایط عملکرد (ابزار، مواد، تجهیزات، زمان، مکان و...)	نتایج ممکن	استاندارد (شاخص‌ها/داوری/نمره دهی)	نمره
حل مسئله تکرار	مکان: کارگاه استاندارد رایانه تجهیزات: کاغذ - نوشت افزار زمان: ۱۵ دقیقه	بالاتر از حد انتظار	تعیین داده، اطلاعات، ورودی و خروجی در مسئله - ارائه راه حل برای مسئله - توسعه راه حل مسئله	۳
		در حد انتظار	تعیین داده، اطلاعات، ورودی و خروجی در مسئله - ارائه راه حل برای مسئله	۲
		پایین تر از حد انتظار	تعیین داده، اطلاعات، ورودی و خروجی در مسئله	۱

## حلقه for

ساختار تکرار for برای پیاده‌سازی حلقه معین مناسب است.  
روندنمای حلقه for



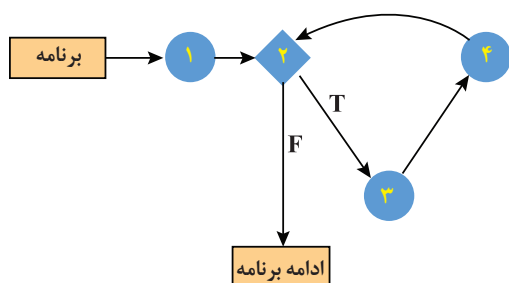
## شکل رایج for در سی شارپ

for (تغییر مقدار متغیر حلقه ; شرط حلقه ; مقداردهی اولیه متغیر)

۱ ۲ ۴

۳

دستور یا دستوراتی که باید در حلقه تکرار شود



شکل ۱- ترتیب اجرای دستورات حلقه for

ابتدا مقداردهی اولیه متغیر انجام شده (۱)، در صورت صحیح بودن شرط حلقه (۲) دستورات بدنه حلقه (۳) اجرا می‌شود، پس از اجرای دستورات بدنه حلقه، مقدار متغیر حلقه تغییر می‌کند (۴) و دوباره شرط حلقه بررسی می‌شود. با نادرست شدن شرط حلقه کنترل برنامه به دستورات پس از حلقه منتقل می‌شود. مطابق شکل ۱ دستورات مراحل ۲، ۳ و ۴ تا برقراری شرط تکرار می‌شوند.

## کارگاه ۱ تبدیل روندنما به برنامه

می‌خواهیم روندنمایی که برای مطب دندانپزشکی طراحی کردیم را به برنامه تبدیل کنیم.

۱ یک پروژه جدید به نام Counter ایجاد کنید.

۲ قطعه کد زیر را در متد Main() وارد کرده، برنامه را اجرا کنید.

```
for(int i=1; i<=20 ; i++)
```

```
Console.WriteLine (i);
```

۳ برنامه را طوری تغییر دهید که برای یک درمانگاه خیریه با تعداد بیماران بیشتر قابل استفاده باشد. شماره‌های مورد نیاز از ۱۰۰ تا ۹۹۹ است.

۴ برنامه را طوری تغییر دهید که حداکثر تعداد بیماران را از ورودی دریافت کند.

## کارگاه ۲ بررسی شرایط حلقه تکرار معین

برای نظافت کارگاه و کلاس، هنجریان یک کلاس ۳۰ نفره را به دو دسته با شماره‌های زوج و فرد تقسیم کرده‌ایم. هنجریانی که در دفتر هنرآموز با شماره زوج ثبت شده‌اند برای نظافت کلاس و هنجریان با شماره فرد برای نظافت کارگاه انتخاب شده‌اند. می‌خواهیم برنامه‌ای بنویسیم که شماره هنجریانی که برای نظافت کلاس انتخاب شده‌اند را به ترتیب در خط‌های جداگانه چاپ کند.

۱ یک پروژه جدید به نام Even ایجاد کنید.

۲ قطعه کد زیر را در متد Main() وارد کرده، خروجی را بررسی کنید

برای سرعت برنامه‌نویسی در VS، پس از نوشتن دستور for دو بار کلید tab را فشار دهید و دستور را ویرایش کنید.

یادداشت



```
for (int i = 2; i <= 30; i += 2)
```

```
Console.Write (" {0,5}", i);
```

۳ پس از پرانتز دستور for علامت ; قرار داده، خروجی را بررسی کنید.

نتیجه را برای هم‌گروهی و هنرآموز خود توضیح دهید.

۴ قطعه کد مرحله ۲ را به صورت زیر تغییر داده، نتیجه را با مرحله قبل مقایسه کنید.

```
for (int i = 2; i <= 30; i++)
```

```
{
```

```
    i++;
```

```
    Console.Write (" {0,5}", i);
```

```
}
```

۵ بررسی کنید در صورتی که آکولادهای قطعه کد بالا حذف شوند خروجی چه تغییری می‌کند؟

۶ قطعه کد مرحله ۲ را طوری تغییر دهید تا شماره هنجریانی که باید کارگاه را نظافت کنند چاپ کند.

۷ در عبارت منطقی، علامت <= را به >= تبدیل کنید، خروجی چه تغییری می‌کند؟





خروجی قطعه کدهای زیر را با کمک هم گروهی خود و بدون استفاده از رایانه بنویسید، سپس آن را اجرا و نتیجه را مقایسه کنید.

حلقه	خروجی برنامه از نظر شما	خروجی برنامه پس از اجرا
<code>for(int i=5; i &gt;=1; i--)</code> <code>Console.WriteLine (i);</code>		
<code>for(int i=0; i&lt;10; i +=3)</code> <code>Console.WriteLine (i);</code>		
<code>for(int i=10; i &gt;=0; i- = 2)</code> <code>Console.WriteLine (i);</code>		

آیا متغیر حلقه می تواند از نوع `byte`، `float` یا `char` باشد؟



هنرآموزی می خواهد ده نفر از هنرجویان یک کلاس را به گروه های ۲ نفره تقسیم کند، او قصد دارد در هر گروه به ترتیب یک نفر از اول فهرست و یک نفر از آخر فهرست را قرار دهد. برنامه ای بنویسید که شماره هنرجویان هر گروه را در خروجی نمایش دهد.



### کارگاه ۳ کار با متغیر خارج بلاک

می خواهیم برنامه ای بنویسیم که بازی هپ را برای مضارب ۵ شبیه سازی کند. در بازی هپ، هر جا که به مضرب عدد تعیین شده می رسیم باید پیام هپ را چاپ کنیم.

۱ یک پروژه جدید به نام `Game` ایجاد کنید.

۲ قطعه کد زیر را در متد `Main()` وارد کرده، خروجی را بررسی کنید.

```
for (int i = 1; i <= 100; i++)
{
    if (i % 5 == 0)
        Console.WriteLine ("Hop");
    else
        Console.WriteLine (i);
}
```

برنامه را اجرا کنید. عملکرد برنامه چیست؟

۳ دستور زیر را به انتهای برنامه اضافه کنید تا مقدار نهایی متغیر `i` پس از اتمام حلقه چاپ شود.

دلیل خطای زیر چیست؟ به کمک هنرآموز خود راه حل آن را پیدا کنید.

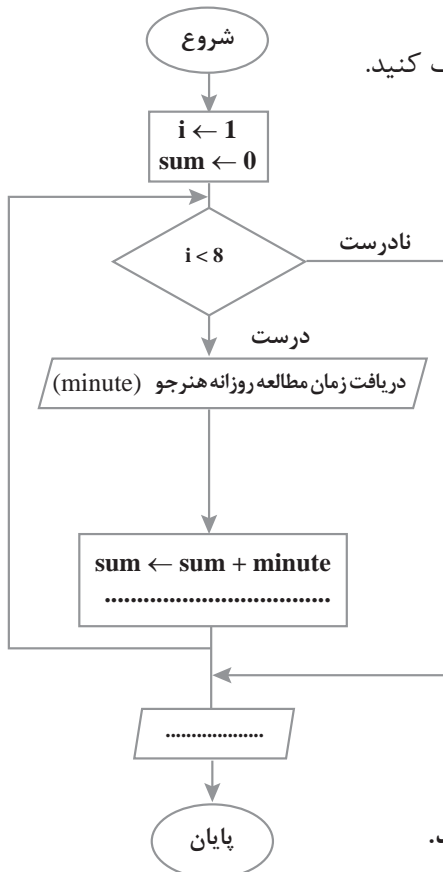
```
Console.WriteLine ("Final i :"+i);
```

The name 'i' does not exist in the current context

۴ برنامه را برای مضارب ۳ بازنویسی کنید.

یک هنرستان می‌خواهد جهت ترویج فرهنگ کتابخوانی مجموع زمان مطالعه هفتگی هر هنرجو را برحسب دقیقه محاسبه کند و در تابلو اعلانات هر کلاس نمایش دهد. برای حل این مسئله، مدت زمان مطالعه هر روز هنرجو را دریافت کرده و با مجموع زمان‌ها جمع می‌کنیم. هفته ۷ روز است بنابراین عمل دریافت مدت زمان مطالعه روزانه و جمع بستن آن با بقیه زمان‌ها ۷ بار تکرار می‌شود. **۱ روندنمای مسئله را کامل کنید.**

پس از تکمیل روندنما، قسمت‌های مربوط به حلقه را پررنگ کنید.



**۲ یک پروژه جدید به نام SumOfStudy ایجاد کنید.**

**۳ قطعه کد زیر را در متد Main() وارد کرده، برنامه را کامل کنید.**

```

.....
for (int i= 1; i<8 ; i++)
{
    Console.WriteLine("Please rate the day {0} enter in minutes=", i);
    minute = int.Parse(Console.ReadLine());
    sum = sum + ..... ;
}
Console.WriteLine("you studied for {0} minutes a week", sum);
    
```

**۴ برنامه را طوری تغییر دهید که مجموع ساعات مطالعه ماهانه هنرجو را نمایش دهد.**

**۵ برنامه را توسعه دهید تا میانگین ساعت مطالعه ماهانه هنرجو را نمایش دهد.**

## کارگاه ۵ | خروج زودرسی از حلقه

می‌خواهیم بازی حدس ماه تولد را بنویسیم. کاربر اول شماره ماه تولد خود را وارد کرده، کاربر دوم باید شماره ماه را حدس بزند. با هر حدس اشتباه یک امتیاز از کاربر دوم کم می‌شود و حداکثر ۵ بار اجازه دارد تا شماره ماه را حدس بزند. در آخر برنامه امتیاز بازیکن دوم نمایش داده می‌شود. امتیاز اولیه ۵ است.

۱ یک پروژه جدید به نام **MonthGuess** ایجاد کنید.

۲ کد زیر را تکمیل کرده، در متد **Main()** وارد کنید و برنامه را اجرا کنید.

در کد زیر متغیرهای موردنیاز برنامه را اعلان کرده، دستورات دریافت آنها از ورودی را تکمیل کنید.

```
.....
int score= 5;
Console.WriteLine ("Enter number your month:");
month = .....
Console.Clear();
for (int i = 1; i <= 5; i++)
{
    Console.WriteLine ("your guess?");
    guess = .....
    if (guess == month)
    {
        Console.ForegroundColor= ConsoleColor.Green;
        Console.WriteLine ("you win");
        Console.WriteLine ("your score: {0}", score);
        break;
    }
    else
    {
        score -- 1;
        Console.WriteLine ("try again !");
    }
}
```

در مواقعی که نیاز دارید برنامه زودتر از موعد از حلقه خارج شود می‌توانید از دستور **break** استفاده کنید.

۳ خروجی برنامه را با مقادیر زیر Trace کنید.

مقدار month را یازده وارد کنید.			
i	guess	score	خروجی
۱	۳		
۲	۸		
۳	۲		
۴	۴		
۵	۵		

مقدار month را پنج وارد کنید.			
i	guess	score	خروجی
۱	۳		
۲	۸		
۳	۵		

۴ برنامه را اصلاح کنید.

پس از اجرای برنامه و Trace متوجه شدید که با حدس اشتباه در بار پنجم مشابه دفعات قبل پیام **try again!** دیده می‌شود، برنامه را طوری تغییر دهید که پس از ۵ بار حدس اشتباه کاربر دوم، پیام مناسب دیگری چاپ شده، شماره ماه تولد را نمایش دهد.

جدول زیر را به کمک هم‌گروهی خود تکمیل کنید.

فعالیت گروهی



خروجی برنامه پس از اجرا	خروجی برنامه از نظر شما	قطعه کد
		<pre>for (int k = 10; k &lt;= 15; ) {     Console.WriteLine(k);     k++; }</pre>
		<pre>int i = 1; Console.ForegroundColor = ConsoleColor.Cyan; for (; i &lt;= 10; i++) {     Console.WriteLine("Iran");     i++; }</pre>
		<pre>Console.ForegroundColor = ConsoleColor.Cyan; for (;) {     Console.WriteLine("Iran"); }</pre>

## فعالیت منزل



برای قطعه کد زیر پس از اجرا و بررسی نتیجه، یک مسئله کاربردی بنویسید.

```
for (int i = 0; i < 100; i += 3)
    Console.WriteLine("{0,3}", i);
```

## فیلم



فیلم شماره ۱۱۱۰۲: اجرای گام به گام برنامه

## فعالیت کارگاهی



– پس از مشاهده فیلم، برنامه MonthGuess را با کمک کلید F10 گام به گام اجرا کنید. تغییرات متغیرها را در جدول زیر یادداشت کنید.

month			
i	guess	score	خروجی

– برنامه کارگاه ۳ را با استفاده از پنجره local گام به گام اجرا کنید. چرا پس از اجرا و اتمام کار حلقه، متغیر i از پنجره local حذف می‌شود؟

## فعالیت منزل



– هنرستان دخترانه پروین اعتصامی هر سال به مناسبت میلاد حضرت معصومه (سلام الله علیها) و روز دختر به هنجریانی که نام آنها معصومه است هدیه می‌دهد. برنامه‌ای طراحی کنید که با دریافت نام هنجریان یک هنرستان تعداد هنجریانی که هم‌نام حضرت معصومه (س) هستند را نمایش دهد. تعداد کل هنجریان هنرستان در ابتدای برنامه از کاربر دریافت شود.

– برنامه را با کمک کلید F10 گام به گام اجرا کرده، با وارد کردن نام ۵ هنجرو مقدار متغیرهای ورودی و خروجی را مشاهده کرده، در جدول زیر وارد کنید.

خروجی	count	name	مقدار متغیر حلقه / i
			۱
			۲
			۳
			۴
			۵

آیا تا به حال بازی حدس عدد را با دوستان خود انجام داده‌اید؟ در این بازی بازیکن اول عددی را در ذهن خود انتخاب کرده و بازیکن دوم عددهایی را حدس می‌زند تا وقتی که عدد را پیدا کند. به نظر شما آیا از ابتدا مشخص است که بازیکن دوم با چند حدس به جواب می‌رسد؟

آیا میزان مسافتی که خودرو با یک باک پر از بنزین طی می‌کند، همیشه یکسان است؟

آیا می‌دانید در زمانی که چراغ راهنمایی یک تقاطع سبز است، چه تعدادی خودرو از تقاطع عبور می‌کند؟

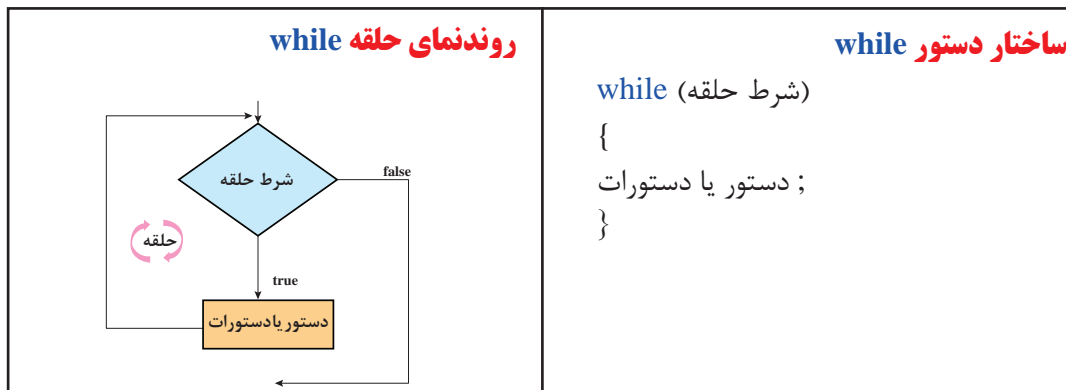
تعیین تعداد خودروهایی که از چهارراه تا زمانی که چراغ راهنمایی سبز است، عبور می‌کنند یک حلقه نامعین است. چون تعداد خودروهایی که می‌توانند در آن زمان از چهارراه عبور کنند مشخص نیست.

آیا مشخص است که یک فرد برای دریافت گواهینامه راهنمایی و رانندگی چند بار آزمون می‌دهد؟

همه مثال‌های بالا یک نوع حلقه نامعین هستند، که از ابتدا تعداد تکرار مشخص نیست. در این گونه موارد اغلب از دستور **while** استفاده می‌شود.

با کمک هم گروهی و هنرآموز خود دو مثال دیگر برای حلقه نامعین بنویسید.

فعالیت گروهی



در صورت درست بودن شرط حلقه، دستور یا دستورات بدنه حلقه اجرا می‌شود و شرط خروج از حلقه **while**، نادرست بودن شرط حلقه است.

### کارگاه ۶ به کارگیری حلقه while

می‌خواهیم میانگین نمره درس ادبیات هنرجویان پایه یازدهم یک هنرستان را محاسبه کنیم. برای این کار باید با دریافت نمره هنرجویان مجموع نمرات را محاسبه کنیم، اما تعداد هنرجویان مشخص نیست. بنابراین شرط پایان عملیات را وارد کردن عدد منفی برای نمره در نظر می‌گیریم.

۱ روندنامی مسئله را رسم کنید.

۲ یک پروژه جدید به نام Average ایجاد کنید.

۳ قطعه کد زیر را در متد Main() وارد کنید و برنامه را تکمیل کنید.

```
float sum=0,avg;
int count=0;
Console.WriteLine("Enter mark {0} =", count+1);
float mark=float.Parse ( Console.ReadLine());
while(.....)
{
    sum = sum + mark;
    count++;
    Console.WriteLine("Enter mark {0} =", count+1);
    mark = float.Parse(Console.ReadLine( ));
}
```

کاربرد متغیر count چیست؟

۴ دستورات لازم برای محاسبه و نمایش میانگین را به برنامه اضافه کنید.

قطعه کدهای زیر را Trace کرده، اولین و آخرین مقدار چاپ شده را در جدول زیر بنویسید.

دستور	اولین مقدار در خروجی	آخرین مقدار در خروجی
<pre>int i = 99; while (i &gt;= 1)     Console.WriteLine (- i);</pre>		
<pre>int i = 100; while (i &lt;= 1000)     Console.WriteLine (++i);</pre>		

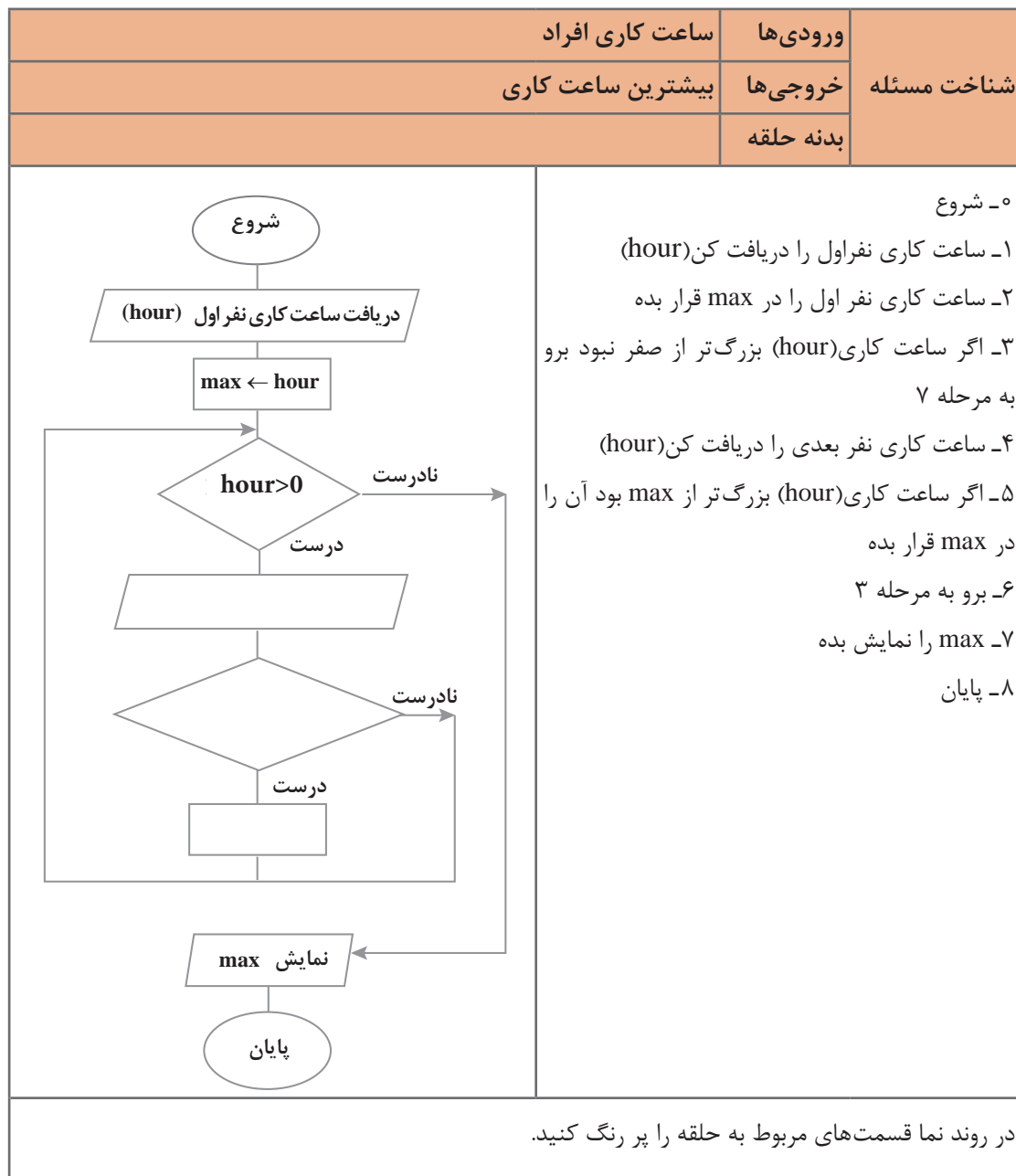
فعالیت کارگاهی



## کارگاه ۷ یافتن بزرگ‌ترین مقدار

یک شرکت خدماتی با هدف تشویق کارمندان خود می‌خواهد به فردی که بیشترین ساعت کاری را در ماه اسفند داشته است، مبلغ ۲۰۰۰۰۰ تومان بن خرید هدیه دهد. می‌خواهیم برنامه‌ای بنویسیم که در انتهای ماه ساعت‌های کاری هر کارمند را دریافت کرده، بیشترین ساعت کاری را چاپ کند.

الگوریتم مسئله به صورت زیر است:





۱ یک پروژه جدید به نام Max ایجاد کنید.

۲ متغیرهای استفاده شده در قطعه کد زیر را اعلان کرده، قطعه کد زیر را تکمیل و برنامه را اجرا کنید.

```
.....
Console.ForegroundColor = ConsoleColor.Blue;
```

```
Console.WriteLine("\nEnter hour");
```

```
.....
hour = .....
```

```
max=hour;
```

```
while (hour >0)
```

```
{
```

```
    if (hour > max)
```

```
        max = hour;
```

```
    Console.WriteLine ("\nEnter 0 for Exit");
```

```
    Console.Write ("Enter hour: ");
```

```
    hour = int.Parse (Console.ReadLine());
```

```
}
```

```
Console.WriteLine("max= {0}", max);
```

شرط خروج از حلقه در این برنامه چیست؟

۳ برنامه را طوری تغییر دهید که کمترین ساعت کاری را نمایش دهد.

۴ برنامه را طوری تغییر دهید که بیشترین و بیشترین ساعت کاری را نمایش دهد.

```
-----Console Calculator-----
1.Enter A , B:
2.CALC A+B=
3.CALC A*B=
4.CALC A/B=
5.CALC A-B=
6.Exit:
-----
A=0    B=0
-----
Enter Number(1-6):
```

ماشین حساب کنسولی همانند شکل روبه‌رو ایجاد کنید.

فعالیت کارگاهی



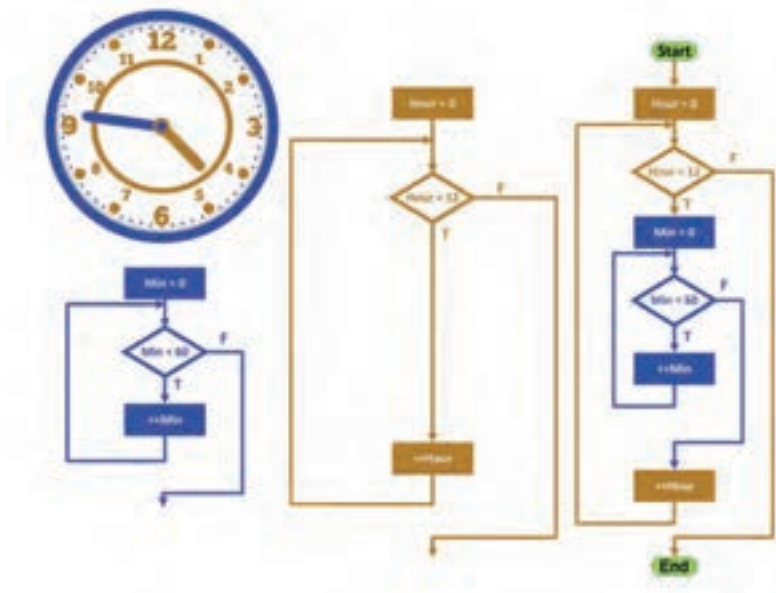
## ارزشیابی مرحله ۲



مرحله کار	شرایط عملکرد (ابزار، مواد، تجهیزات، زمان، مکان و ...)	نتایج ممکن	استاندارد (شاخص‌ها/داوری/نمره دهی)	نمره
ایجاد برنامه با حلقه	مکان: کارگاه استاندارد رایانه تجهیزات: رایانه‌ای که نرم‌افزار IDE برنامه‌نویسی روی آن نصب است زمان: ۲۰ دقیقه	بالاتر از حد انتظار	تعیین اجزای ساختار حلقه با توجه به مسئله و نوشتن کد آن - trace کردن حلقه در برنامه - رفع خطای برنامه	۳
		در حد انتظار	تعیین اجزای ساختار حلقه با توجه به مسئله و نوشتن کد آن	۲
		پایین‌تر از حد انتظار	تعیین اجزای ساختار حلقه با توجه به مسئله	۱

## حلقه‌های متداخل

در برخی از مواقع، یک دستور تکرار را در داخل دستور تکرار دیگری به کار می‌بریم. به عبارت دیگر هنگامی که در داخل یک حلقه، حلقه دیگری قرار داشته باشد، حلقه‌های تودرتو یا متداخل نامیده می‌شوند. دبیر ورزش از هنجریان خود خواسته است تا پنج بار دور حیاط مدرسه دویده، در پایان هر دور سه حرکت بارفیکس انجام دهند. می‌خواهیم در یک مجتمع ۱۰ طبقه‌ای که هر طبقه دارای ۸ واحد است، غذای نذری پخش کنیم. چند غذا برای این مجتمع نیاز است؟ همه این مثال‌ها کارهای تکراری است که در درون کار تکراری دیگری قرار دارند و کاربرد حلقه‌های متداخل را بیان می‌کنند.



به کمک هم گروهی خود شکل را توصیف کنید.

فعالیت گروهی



## کارگاه ۸ کاربرد حلقه‌های متداخل

می‌خواهیم برای دانش‌آموزان کلاس سوم برنامه‌ای بنویسیم که جدول ضرب ۵ در ۵ را نمایش دهد.

۱ یک پروژه جدید به نام **Multiple** ایجاد کنید.

۲ قطعه کد زیر را در متد **Main()** وارد کنید.

```
for (int i = 1; i <= 5; i++)
{
    Console.Write (" {0,4}", i * 1);
    Console.Write (" {0,4}", i * 2);
    Console.Write (" {0,4}", i * 3);
    Console.Write (" {0,4}", i * 4);
    Console.Write (" {0,4}", i * 5);
    Console.WriteLine();
}
```

برنامه را اجرا کنید. آیا خروجی یک جدول ضرب را نشان می‌دهد؟

۳ با توجه به اعداد مشخص شده یک تا پنج، در کد مرحله ۲ با استفاده از دستور حلقه **for** دستورهای **Console.Write** را به یک دستور تبدیل کنید.

۴ برای تنظیم نمایش جدول ضرب به جای جا نگهدار از دستور **SetCursorPosition** استفاده کنید.

۵ برنامه را با کمک کلید **F10** خط به خط اجرا کنید.

سیستم نوبت‌دهی مطب دندانپزشکی را طوری تغییر دهید که برای یک هفته که شامل شش روز کاری است نوبت‌دهی انجام دهد.

فعالیت منزل



## کارگاه ۹ توسعه حلقه متداخل

تعداد اسکناس ۵ هزار تومانی	تعداد اسکناس ۲ هزار تومانی
۸	۵
	۱۰

آیا به خودپرداز جهت دریافت وجه نقد مراجعه کرده‌اید؟ فکر می‌کنید خودپرداز چگونه پرداخت انواع اسکناس را انجام می‌دهد؟ اگر خودپردازی دارای اسکناس‌های ۲ و ۵ هزار تومانی باشد و شما درخواست ۵۰ هزار تومان وجه نقد کنید، چگونه پرداخت را انجام می‌دهد؟ چه روش‌های پرداختی می‌تواند داشته باشد؟ جدول رو به رو را تکمیل کنید.

خودپرداز از شمارش برای پرداخت وجه نقد استفاده می‌کند و با توجه به موجودی اسکناس‌ها پرداخت را تعیین می‌کند.

**الگوریتم:** اگر X تعداد اسکناس‌های ۲ هزار تومانی و Y تعداد اسکناس‌های ۵ هزار تومانی باشد برای اینکه پرداخت معادل ۵۰ هزار تومانی باشد شرط روبه‌رو باید برقرار باشد:  $2000X + 5000Y = 50000$   
 به کمک حلقه متداخل همه مقادیر X و Y را بررسی می‌کنیم هر کدام در شرط بالا صدق کرد، می‌تواند یک پرداخت برای ۵۰ هزار تومانی باشد.

۱ یک پروژه به نام ATM ایجاد کنید.

۲ قطعه کد زیر را در بخش Main بنویسید.

```
for (int X = 0; X <= 25; X++)
{
    for(int Y = 0; Y <= 10; Y++)
    {
        if (2000 * X + 5000 * Y == 50000)
            Console.WriteLine ("X={0},Y={1}", X, Y);
    }
}
```

برنامه را اجرا کنید. آیا با جدولی که شما کامل کرده‌اید مطابقت دارد؟ فکر می‌کنید خودپرداز کدام پرداخت را انجام می‌دهد؟

چرا مقدار نهایی X مقدار ۲۵ و مقدار نهایی Y مقدار ۱۰ در نظر گرفته شده است؟

۳ برنامه چند حالت را با مقادیر مختلف (X,Y) آزمایش می‌کند تا به جواب برسد؟

۴ برنامه را با فرض اینکه خودپرداز اسکناس ۱۰ هزار تومانی هم دارد، بنویسید.

۵ برنامه را طوری تغییر دهید که وجه نقد درخواستی کاربر را از ورودی دریافت کنیم.

- برنامه کارگاه ۶ را توسعه دهید تا با دریافت نمرات بقیه دروس پایه یازدهم، میانگین آنها را محاسبه و نمایش دهد.  
 - کمترین و بیشترین میانگین نمره را نمایش دهید.

فعالیت منزل



## ارزشیابی مرحله ۳

مرحله کار	شرایط عملکرد (ابزار، مواد، تجهیزات، زمان، مکان و ...)	نتایج ممکن	استاندارد (شاخص‌ها/داوری/نمره دهی)	نمره
ایجاد برنامه با حلقه‌های متداخل	مکان: کارگاه استاندارد رایانه تجهیزات: رایانه‌ای که نرم‌افزار IDE برنامه‌نویسی روی آن نصب است زمان: ۲۰ دقیقه	بالاتر از حد انتظار	نوشتن برنامه - اجرای گام به گام حلقه متداخل - رفع خطای حلقه متداخل	۳
		در حد انتظار	نوشتن برنامه	۲
		پایین‌تر از حد انتظار	تعیین تعداد حلقه‌های یک مسئله	۱

**معیار شایستگی انجام کار:**

کسب حداقل نمره ۲ از مرحله ایجاد برنامه با حلقه  
 کسب حداقل نمره ۲ از بخش شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش  
 کسب حداقل میانگین ۲ از مراحل کار



## جدول ارزشیابی پایانی

شرح کار:

۱ حل مسئله تکرار

۲ ایجاد برنامه با حلقه

۳ ایجاد برنامه با حلقه‌های متداخل

استاندارد عملکرد:

با استفاده از دانش ساختار تکرار در برنامه‌نویسی، مسئله را تحلیل و در صورت نیاز برنامه را با استفاده از ساختار تکرار کدنویسی و خطایابی کند.

شاخص‌ها:

شماره مرحله کار	شاخص‌های مرحله کار
۱	تعیین داده، اطلاعات، ورودی و خروجی در مسئله - ارائه راه‌حل برای مسئله - توسعه راه حل در صورت نیاز
۲	تعیین و مقداردهی اجزای ساختار تکرار- نوشتن برنامه - تبدیل راه حل مسئله تکرار به برنامه - خطایابی و رفع خطاهای برنامه
۳	نوشتن برنامه - رفع خطاهای برنامه

شرایط انجام کار و ابزار و تجهیزات:

مکان: کارگاه رایانه مطابق استاندارد تجهیزات هنرستان‌ها

تجهیزات: رایانه‌ای که نرم‌افزار IDE برنامه‌نویسی روی آن نصب است - نوشتافزار - کاغذ

زمان: ۶۰ دقیقه (حل مسئله تکرار ۱۵ دقیقه - ایجاد برنامه با حلقه ۲۰ دقیقه - ایجاد برنامه با حلقه‌های متداخل ۲۰ دقیقه)

معیار شایستگی:

ردیف	مرحله کار	حداقل نمره قبولی از ۳	نمره هنرجو
۱	حل مسئله تکرار	۱	
۲	ایجاد برنامه با حلقه	۲	
۳	ایجاد برنامه با حلقه‌های متداخل	۱	
<div>شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش:</div> <div>حل مسئله، شناسایی مسئله، تولید/ ارزیابی راه‌حل‌ها - زبان فنی</div> <div>رعایت ارگونومی</div> <div>حفاظت از تجهیزات کارگاه</div> <div>دقت در تعیین بدنه حلقه مورد نیاز برای برنامه - استفاده از ابزارهای خطایابی برای رفع خطای برنامه</div>		۲	
میانگین نمرات			*

\* حداقل میانگین نمرات هنرجو برای قبولی و کسب شایستگی، ۲ است.

## واحد یادگیری ۴

### ■ شایستگی کار با آرایه

#### آیا تا به حال پی برده اید

- اسامی هنرجویان یک کلاس و نمرات آنها چگونه در حافظه نگه‌داری و پردازش می‌شود؟
- چگونه می‌توان مجموعه‌ای از واژه‌ها و معانی آنها را بر اساس حروف الفبا مرتب کرد؟
- چگونه می‌توان معنی یک واژه را در بین واژه‌های مرتب‌شده بر اساس حروف الفبا جست‌وجو کرد؟
- چگونه می‌توان نمرات هنرجویان را همراه با نام آنها به‌صورت نزولی مرتب کرد؟

هدف از این واحد شایستگی استفاده از آرایه است.

#### استاندارد عملکرد

با استفاده از دانش ساختار آرایه مسئله را تحلیل کرده، برنامه را ایجاد و خطایابی کند.

می‌دانیم برای نگه‌داری داده‌ها در حافظه، از متغیر استفاده می‌شود. برای مثال برای نگه‌داری نام یک هنرجو، متغیری از نوع رشته تعریف می‌شود.

سعید
مجید
محمد
علی
حسین

فهرست اسامی هنرجویان کلاس را در نظر بگیرید. اگر در برنامه‌ای نیاز باشد که نام تمام هم‌کلاسی‌های خود را دریافت و نگه‌داری کنید، چه خواهید کرد؟ آیا به تعداد هم‌کلاسی‌هایتان متغیر تعریف می‌کنید؟ در این مثال، به جای اینکه اسامی هر یک از هنرجویان را در متغیرهای جدا ذخیره کنیم، می‌توانیم جدولی به شکل روبه‌رو تعریف کرده، اسامی را در آن نگه‌داری کنیم:

شماره	نام
۱	سعید
۲	مجید
۳	محمد
۴	علی
۵	حسین

این جدول شامل نام پنج هنرجو است. اسامی هنرجویان در ردیف‌های مجزا نوشته شده است. برای این جدول، یک نام مناسب انتخاب می‌کنیم. همچنین برای دسترسی به نام هنرجویان، در کنار هر نام، یک شماره قرار می‌دهیم. برای مثال، نام نفر سوم، محمد است.



- روندنمایی رسم کنید که اسامی پنج هنرجو را دریافت کرده، سپس اسامی را از آخر به اول نمایش دهد.  
**روش انجام کار:** اگر برای هر هنرجو متغیر جداگانه‌ای تعریف کنیم، نیاز به تعریف پنج متغیر است. دریافت مقدار از ورودی و قرار دادن در متغیر باید برای هر پنج اسم انجام شود؛ یعنی نام اول را دریافت و در متغیر اول قرار دهیم، سپس نام دوم را دریافت و در متغیر دوم قرار دهیم و این روال را تا متغیر پنجم ادامه دهیم. در پایان متغیر پنجم، چهارم، سوم، دوم و اول را نمایش دهیم.  
 - برنامه این الگوریتم را به زبان سی شارپ بنویسید.

در این مثال، اگر تعداد اسامی ۳۰ نفر باشد:

چه تعداد دستور خواندن از ورودی و نوشتن روی صفحه نمایش لازم است؟

آیا می‌توان از دستور for برای خواندن ورودی‌ها و نوشتن اسامی استفاده کرد؟

برای چنین مسئله‌هایی که با داده‌های زیاد سروکار دارند، چه روشی را پیشنهاد می‌کنید؟

در زبان‌های برنامه‌نویسی مانند سی شارپ برای نگه‌داری تعدادی داده هم نوع، از مفهوم آرایه استفاده می‌شود.

**آرایه**، فهرستی از عناصر هم نوع است که تحت یک نام مشترک کنار هم در حافظه ذخیره می‌شوند. هر خانه آرایه، یک **عنصر آرایه** نامیده می‌شود. برای تفکیک و دسترسی به هر عنصر، از یک عدد صحیح به نام **اندیس** استفاده می‌شود. در زبان سی شارپ، اندیس اولین عنصر آرایه صفر است، اندیس عنصر بعدی یک و به همین ترتیب اندیس عناصر یکی یکی اضافه می‌شوند.

فهرست اسامی ۵ هنرجو را می‌توان به شکل زیر در نظر گرفت:

آخرین اندیس → ۴	۳	۲	۱	۰ ← اولین اندیس
تعداد عناصر ۵ عدد است	حسین	علی	محمد	مجید
				عنصر

تعداد عناصر آرایه را طول آرایه می‌نامند. آرایه بالا دارای طول ۵ است و عناصر آن از ۰ تا ۴ شماره‌گذاری شده است.

## اعلان آرایه

به یاد دارید برای اعلان یک متغیر از نوع رشته‌ای به شکل زیر عمل می‌کردیم:

`string name;`

در زبان سی شارپ می‌توان تعریف و ایجاد آرایه را در دو مرحله انجام داد. برای تعریف یک آرایه که از نوع رشته‌ای است، در مرحله اول، مشابه اعلان متغیر رشته‌ای به شکل زیر عمل می‌شود:

`string[] name;`

در این دستور متغیر `name`، آرایه‌ای از نوع رشته‌ای اعلان شده است؛ اما تعداد خانه‌های آرایه مشخص نیست و حافظه‌ای برای آرایه تخصیص نیافته است.

در مرحله دوم با استفاده از عملگر **new** و مشخص کردن اندازه آرایه، حافظه مناسب به آرایه اختصاص داده شده، آرایه ایجاد می‌شود.

`name = new string[5];`

در این دستور، `name` نام آرایه‌ای است که در مرحله قبل اعلان شده بود. `string` نوع عناصر و ۵ تعداد عناصر آرایه است. اندیس عناصر آرایه از ۰ تا ۴ است.

فعالیت کارگاهی



یک پروژه جدید ایجاد کنید و دستور اعلان آرایه و دستور تخصیص حافظه به آرایه `name` را بنویسید. بررسی کنید محتوای هر عنصر آرایه چیست؟

```
string[] name;
name = new string[5];
name {string[5]}
```

برای دیدن محتوای عناصر آرایه، برنامه را با کلید F10 اجرا کنید. سپس با قرار دادن اشاره‌گر ماوس روی نام آرایه و کلیک روی علامت + مقادیر آرایه را مشاهده کنید.



مرحله اعلان آرایه و تخصیص حافظه به آن را می‌توانیم در یک دستور و به‌صورت زیر بنویسیم:

```
string[] name = new string[5];
```

### یکی از اشکال تعریف و ایجاد آرایه

[تعداد عناصر آرایه] نوع داده = new نام آرایه [نوع داده];

فعالیت کارگاهی



- آرایه‌ای به نام salary برای نگهداری حقوق ۳۰۰ کارمند اعلان کنید.
- آرایه‌ای به نام vowels برای نگهداری حروف صدا دار انگلیسی اعلان کنید.
- آرایه‌ای به نام lamp برای نگهداری وضعیت روشن و خاموش بودن پنج لامپ اعلان کنید.
- آرایه‌ای با نام دلخواه برای نگهداری اسامی ماه‌های سال اعلان کنید.
- آرایه‌ای با نام دلخواه برای نگهداری معدل هنرجویان کلاس اعلان کنید.

### مقداردهی عناصر آرایه

در فهرست اسامی هنرجویان، با دانستن شماره هنرجو در فهرست اسامی، به راحتی می‌توان به نام وی دسترسی داشت.

برای دسترسی به خانه‌های آرایه، از نام آرایه به همراه اندیس عنصر به‌صورت زیر استفاده می‌کنیم:

[اندیس] نام آرایه

برای مثال name[0] اولین عنصر آرایه و name[2] سومین عنصر آرایه است.

هر متغیر در هر لحظه می‌تواند فقط یک مقدار داشته باشد و با انتساب مقدار جدید، مقدار قبلی آن از بین می‌رود. در آرایه نیز با انتساب یک مقدار جدید در هر عنصر، مقدار قبلی از بین رفته، مقدار جدید جایگزین خواهد شد.

برای مقداردهی عناصر آرایه روش‌های مختلفی وجود دارد. یک روش مقداردهی، استفاده از دستور انتساب است. برای مثال دستور زیر مقدار maryam را در خانه سوم آرایه رشته‌ای name قرار می‌دهد:

```
name[2] = "maryam";
```

کنجکاوی



چرا برای مراجعه به خانه سوم آرایه، از اندیس ۲ استفاده شده است؟

### مقداردهی عناصر آرایه با دستور انتساب

مقدار = [اندیس] نام آرایه

برداشت



آنچه آموختیم:

۱. ....
۲. ....
۳. ....

## کارگاه ۱ تعریف و مقداردهی آرایه

هنرجویی قصد دارد که قطعات رایانه را جداگانه خریداری کرده، در منزل آن را مونتاژ کند. می‌خواهیم برنامه‌ای بنویسیم که یک آرایه اعلان کرده، اسامی قطعات خریداری‌شده را در آن قرار دهیم.

۱ پروژه جدیدی به نام **Computer** ایجاد کنید.

۲ کد زیر را در متد **Main()** بنویسید.

```
string[] computer = new string[5];
```

آرایه **computer** دارای ۵ خانه از نوع رشته‌ای است.

۳ اسامی قطعات رایانه را در خانه‌های اول، دوم، سوم، چهارم و پنجم آرایه قرار دهید.

```
computer[0] = "CPU";
```

```
computer[1] = "MainBoard";
```

```
computer[2] = "RAM";
```

```
computer[3] = "HDD";
```

```
computer[4] = "DVD";
```

۴ برای ذخیره کردن اسامی قطعات دیگر، چه تغییری در کد باید بدهیم؟

```
static void Main(string[] args)
{
    string[] computer = new string[5];
    computer[0] = "CPU";
    computer[1] = "MainBoard";
    computer[2] = "RAM";
    computer[3] = "HDD";
    computer[4] = "DVD";
}
```

شکل ۳- مشاهده محتوای عناصر آرایه در زمان اجرا

۵ برنامه را با فشردن کلید **F10** یا از طریق

منوی **Debug** گزینه **Step Over** اجرا کنید.

مشاهده خواهید کرد با هر بار فشردن کلید **F10**،

یک دستور با رنگ زرد مشخص می‌شود. این

دستور، دستوری است که اجرا خواهد شد. برنامه را

تا دستور **computer[2] = "RAM";** اجرا کرده،

اشاره‌گر ماوس را روی دستور قرار دهید. در کادر

ظاهر شده، روی علامت + کنار نام آرایه کلیک

کنید. شکل ۳ مشاهده خواهد شد.

چرا مقدار برخی از عناصر برابر **null** است؟

۶ برای نگه‌داری قیمت قطعات رایانه، آرایه **price** را متناظر با آرایه نام قطعات اعلان کنید.

۷ عناصر آرایه **price** را مقداردهی کنید.

کد زیر را تکمیل کرده، به متد **Main** اضافه کنید.

```
price[0] = 200000;
```

```
price[1] = 150000;
```

```
price[2] = .....;
```

```
.....
```

```
.....
```

۸ برنامه را با کلید **F10** تا رسیدن به دستور مقداردهی **price[2]** اجرا کنید و محتوای آرایه **price** را مشاهده کنید.

محتوای آرایه **price** چیست؟ چرا؟

#### ۹ دستور زیر را جایگزین کد مرحله ۲ و ۳ کنید.

```
string[] computer= new string[] {"CPU", "MainBoard", "RAM", "HDD", "DVD"};
```

روش دیگر مقداردهی عناصر آرایه، مقداردهی در هنگام اعلان و تخصیص حافظه به آرایه است در این حالت باید مقادیر اولیه آرایه مشخص باشد. در این روش آرایه مانند قبل اعلان می‌شود ولی در انتهای اعلان آرایه و در بین علامت‌های { } مقدار عناصر آرایه به ترتیب معین می‌شود.

#### روش دیگر اعلان و مقداردهی آرایه در یک دستور

{ مقدار آخرین عنصر, ... مقدار دوم, مقدار اول } [] نوع داده new = نام آرایه [] نوع داده

#### ۱۰ دستور زیر را جایگزین کد مرحله ۹ کنید.

```
string[] computer= new string[5] {"CPU", "MainBoard", "RAM", "HDD", "DVD"};
```

به جای عدد ۵ عدد ۴ را قرار دهید. خطای رخ داده چیست؟

#### ۱۱ دستور زیر را جایگزین کد مرحله ۹ کنید.

```
string[] computer= {"CPU", "MainBoard", "RAM", "HDD", "DVD"};
```

در حالتی که آرایه دارای مقادیر اولیه مشخص است، می‌توان بدون استفاده از عملگر new آرایه را به صورت زیر ایجاد کرد:

#### اعلان و مقداردهی آرایه بدون استفاده از عملگر new

{ مقدار آخرین عنصر, ... مقدار دوم, مقدار اول } = نام آرایه [] نوع داده

برنامه را اجرا کرده، نتیجه را با مراحل قبل مقایسه کنید.

#### ۱۲ دستوری برای تغییر مقدار عنصر شماره ۳ آرایه computer به "Keyboard" به کد اضافه کنید.

برنامه را با F10 اجرا کرده، محتوای عناصر آرایه computer را مشاهده کنید

#### ۱۲ دستوری بنویسید که قیمت Keyboard را از ورودی دریافت کرده و در عنصر شماره ۳ آرایه price قرار دهد.

برای مقداردهی عناصر آرایه در دستور اعلان آرایه، برای جلوگیری از خطای حاصل از کم یا زیاد بودن تعداد مقادیر انتسابی به آرایه، بهتر است از تعریف و مقداردهی آرایه بدون استفاده از عملگر new استفاده کنیم.

یادداشت



برداشت



#### آنچه آموختیم:

۱. ....
۲. ....
۳. ....

## جدول ارزشیابی شایستگی‌های غیر فنی، ایمنی، بهداشت و توجهات زیست محیطی



شایستگی‌ها	شرایط عملکرد (ابزار، مواد، تجهیزات، زمان، مکان و ...)	نتایج ممکن	استاندارد (شاخص‌ها/داوری/نمره دهی)	نمره
شایستگی‌های غیر فنی	تصمیم‌گیری، تشخیص اهداف و محدودیت‌ها - زبان فنی	قابل قبول	تعیین داده‌ها و پردازش‌هایی که باید روی آنها انجام شود - بررسی لزوم استفاده از آرایه در برنامه	۲
ایمنی و بهداشت	رعایت ارگونومی			
توجهات زیست محیطی	حفاظت از تجهیزات کارگاه	غیر قابل قبول	توجه به ایمنی و بهداشت محیط کارگاه	۱
نگرش	دقت در تشخیص ساختار مناسب داده هنگام تعریف آرایه در برنامه			

● این شایستگی‌ها در ارزشیابی پایانی واحد یادگیری باید مورد توجه قرار گیرند.

## ارزشیابی مرحله ۱



مراحل کار	شرایط عملکرد (ابزار، مواد، تجهیزات، زمان، مکان و ...)	نتایج ممکن	استاندارد (شاخص‌ها/داوری/نمره دهی)	نمره
اعلان آرایه	مکان: کارگاه استاندارد رایانه تجهیزات: رایانه‌ای که نرم‌افزار IDE روی آن نصب است زمان: ۱۰ دقیقه	بالاتر از حد انتظار	ایجاد آرایه و مقداردهی اولیه عناصر آن - رفع خطای برنامه	۳
		در حد انتظار	ایجاد آرایه و مقداردهی اولیه عناصر آن	۲
		پایین‌تر از حد انتظار	تعیین نوع آرایه با توجه به داده‌ها	۱

## نمایش عناصر آرایه

برای نمایش محتوای عناصر آرایه می‌توان از متدهای `Write()` یا `WriteLine()` استفاده کرد. برای مثال برای نمایش محتوای عناصر اول و دوم آرایه `price` خواهیم داشت:

```
Console.WriteLine(price[0]);
Console.Write(price[1]);
```

آیا می‌توان با یک متد `WriteLine()` تمام عناصر آرایه را نمایش داد؟  
حاصل اجرای دستور زیر چیست؟

```
Console.WriteLine(price);
```

کنجکاوی



## کارگاه ۲ نمایش عناصر آرایه

می‌خواهیم برنامه‌ای بنویسیم که با دریافت شماره یک روز هفته، نام آن را نمایش دهد.

۱ پروژه جدیدی به نام `Week` ایجاد کنید.

۲ اعلان آرایه `week` را تکمیل کنید.

```
string[] week = { "saturday", ....., "friday" };
```

آرایه `week` دارای ۷ عنصر از نوع رشته‌ای برای نگه‌داری اسامی روزهای هفته است.

۳ اعلان آرایه `week` و کد زیر را در متد `Main` بنویسید.

```
Console.Write ("Enter the number one day of week (0 - 6): ");
```

```
byte num = byte.Parse (Console.ReadLine());
```

```
Console.WriteLine (week[num]);
```

برنامه را با اعداد مختلف اجرا کنید.

۴ برنامه را با عدد ۷ اجرا کنید.

چه خطایی رخ می‌دهد؟ علت آن چیست؟

۵ اگر بخواهیم به جای اعداد ۰ تا ۶، اعداد ۱ تا ۷ دریافت شود، چه تغییری باید در برنامه ایجاد کنیم؟

۶ دستوراتی به برنامه اضافه کنید که اگر شماره وارد شده در محدوده مجاز نبود، پیام مناسب نمایش داده شود.

یادداشت



دسترسی به عنصری از آرایه که وجود ندارد، باعث بروز خطا می‌شود.

گاهی لازم است به بخشی یا تمامی عناصر آرایه مراجعه شود. برای مثال معلم هنگام حضور و غیاب، اسامی را از ابتدا تا انتهای فهرست هنرجویان می‌خواند. به این عمل پیمایش یا مرور آرایه می‌گوییم. فرض کنید می‌خواهیم اسامی روزهای هفته را نمایش دهیم. یک روش آن است که عناصر آرایه را از ابتدا تا انتها یکی‌یکی نمایش دهیم. به دستورات زیر توجه کنید:

```
string[] week = { "saturday", ....., "friday" };
```

```
Console.WriteLine (week[0]);
```

```
Console.WriteLine (week[1]);
```

```
Console.WriteLine (week[2]);
```

```
Console.WriteLine (week[3]);
```

```
Console.WriteLine (week[4]);
```

```
Console.WriteLine (week[5]);
```

```
Console.WriteLine (week[6]);
```

در این روش باید به تعداد عناصر آرایه از متد WriteLine استفاده شود. اگر تعداد عناصر آرایه زیاد باشد و بخواهیم تمام عناصر آن را پیمایش کنیم، چه روشی پیشنهاد می‌کنید؟ مقادیر خانه‌های یک آرایه دارای ۱۰۰ عنصر را چگونه نمایش می‌دهید؟

برای پیمایش آرایه‌های بزرگ، از حلقه تکرار استفاده می‌شود. برای مثال دستورات نمایش اسامی روزهای هفته را می‌توان به صورت زیر نوشت:

```
for(int i = 0; i < 7; i++)
```

```
    Console.WriteLine(week[i]);
```

برنامه‌ای بنویسید که اسامی روزهای زوج آرایه week را نمایش دهد.

فعالیت کارگاهی



## کارگاه ۳ | پیمایش آرایه

می‌خواهیم برنامه‌ای بنویسیم که قیمت قطعات رایانه را از ورودی دریافت کند و قیمت کل رایانه را به همراه نام و قیمت قطعات در خروجی نمایش دهد.

۱ پروژه جدیدی به نام **PriceOfComputer** ایجاد کنید.

۲ تعداد قطعات رایانه را از ورودی دریافت کنید و دو آرایه با نام‌های **computer** و **price** برای قطعات و قیمت آنها اعلان کنید.

۳ قطعه کد زیر را برای دریافت عناصر آرایه **computer** به برنامه اضافه کنید.

```
for (int i = 0; i < computer.Length; i++)
{
    Console.WriteLine("Enter the name of a computer part");
    computer[i] = Console.ReadLine();
}
```

هر آرایه دارای ویژگی **Length** است که تعداد عناصر آرایه را مشخص می‌کند. **computer.Length** تعداد عناصر آرایه **computer** است.

۴ کد مرحله قبل را تغییر دهید تا علاوه بر عناصر **computer**، قیمت قطعات را هم دریافت کنید.

۵ کد محاسبه قیمت رایانه را بنویسید.

برای محاسبه قیمت کل قطعات رایانه از متغیر **sumPrice** استفاده می‌کنیم:

```
long sumPrice = 0;
for (int i = 0; i < price.Length; i++)
    sumPrice += price[i];
```

۶ برنامه را طوری تکمیل کنید که نام، قیمت قطعات و قیمت کل را نمایش دهد.

رشته، مجموعه متوالی (sequential collection) از نویسه‌ها است و برای نمایش نویسه‌های یک رشته می‌توان از نام رشته و شماره نویسه استفاده کرد. شماره نویسه‌ها از صفر شروع می‌شود. می‌خواهیم برنامه‌ای بنویسیم که یک مقدار رشته‌ای از ورودی گرفته، نویسه‌های آن را جدا کرده، هر کدام را در یک خط بنویسد و معکوس رشته را نمایش دهد.

۱ پروژه جدیدی به نام `StringChar` ایجاد کنید.

۲ دستورات لازم را برای دریافت یک رشته از کاربر بنویسید.

نام متغیر رشته‌ای را `myStr` در نظر بگیرید.

۳ دستورات زیر را به کد اضافه کنید.

```
for (int i = 0; i < myStr.Length; i++)
```

```
    Console.WriteLine(myStr[i]);
```

ویژگی `Length` طول متغیر رشته‌ای یعنی تعداد نویسه‌های رشته را تعیین می‌کند.

برنامه را اجرا کنید. خروجی آن چیست؟

۴ کد زیر را جایگزین مرحله ۳ کنید.

```
foreach (char c in myStr)
```

```
    Console.WriteLine(c);
```

برنامه را اجرا کنید. نتیجه و کد را با مرحله ۳ مقایسه کنید.

اگر بخواهیم به محتوای تمام عناصر آرایه صرف‌نظر از طول آرایه دسترسی داشته باشیم و آرایه را پیمایش کنیم، می‌توان از دستور `foreach` استفاده کرد.

(نام آرایه `in` متغیر حلقه نوع داده) `foreach`

شکل کلی دستور `foreach`

```
{
    دستور;
}
```

در این دستور از اندیس استفاده نمی‌شود و متغیر حلقه، نقش یک عنصر از آرایه را به عهده دارد؛ بنابراین باید از نوع عناصر آرایه باشد.

پس از کلیدواژه `foreach` در VS دو بار کلید `Tab` را فشار دهید تا شکل کامل دستور نوشته شود.

یادداشت





۵ برای نمایش معکوس رشته، کد زیر را تکمیل کنید.

```
for (int i = myStr.Length - 1; i >= ..... ; .....)
```

.....

آیا می‌توان این کد را با دستور `foreach` نوشت؟

- در دستور `foreach` منظور از متغیر حلقه چیست؟  
- آیا می‌توان به‌وسیله عملگر انتساب (=) مقداری را داخل متغیر حلقه `foreach` قرار داد؟

برنامه‌ای بنویسید که سطح شایستگی هنرجویان یک کلاس را به صورت عددی دریافت کرده، به‌وسیله دستور `foreach` تعداد هنرجویانی که سطح شایستگی لازم را کسب نکرده‌اند یعنی دارای سطح شایستگی ۱ هستند نمایش دهد. بررسی شود که سطح شایستگی دریافت شده یکی از سه سطح ۱، ۲ و ۳ باشد.

کنجکاوی



فعالیت منزل



## عملیات روی آرایه

آرایه یک نوع داده است که می‌تواند تعداد زیادی داده هم‌نوع را نگهداری کند. به‌وسیله حلقه‌های تکرار می‌توانیم عملیات یکسانی روی تمام یا برخی از عناصر آرایه انجام دهیم.

### کارگاه ۵ ویرایش عناصر آرایه

می‌خواهیم قیمت محصولات یک فروشنده را از ورودی دریافت کرده، در آرایه ذخیره کنیم. سپس مبلغ مالیات ارزش‌افزوده (VAT) را به قیمت محصولات اضافه کرده، ضمن نمایش قیمت جدید هر محصول، آن را در همان خانه آرایه ذخیره کنیم. تعداد محصولات در ابتدا مشخص نیست اما حداکثر ۵۰ عدد است. با ورود عدد صفر یا عدد منفی به‌عنوان قیمت محصول، دریافت داده پایان می‌یابد.

۱ پروژه جدیدی به نام `ValueAddedTax` ایجاد کنید.

۲ با اعلان آرایه و متغیرهای مناسب، کد زیر را برای دریافت قیمت محصولات تکمیل کنید.

```
for (i = 0; i < 50; i++)
{
    Console.WriteLine("Enter the price of {0} the product: ", i+1);
    temp = int.Parse(Console.ReadLine());
    if (temp > 0)
        priceList[i] = temp;
    else
        break;
}
```

به وسیله این حلقه، حداکثر ۵۰ عدد به عنوان قیمت محصول دریافت می شود. عدد دریافتی ابتدا بررسی می شود. اگر مثبت بود در آرایه قرار می گیرد و اگر صفر یا منفی بود، دستور **break** باعث خروج از حلقه می شود. تعداد قیمت محصولات وارد شده، به وسیله متغیر **i** نگهداری می شود.

۳ دستورات محاسبه مالیات ارزش افزوده را بنویسید.

```
Console.WriteLine ("Product price plus VAT!");
for (int j = 0; j < i; j++)
{
    priceList[j] = (int)(priceList[j] + 0.09 * priceList[j]);
    Console.WriteLine ("Product{0}: {1}", j, priceList[j]);
}
```

در این کد نقش متغیر **i** و **j** چیست؟  
در این قطعه کد با نمایش پیام مناسب، مالیات ارزش افزوده ۹ درصد به قیمت هر محصول اضافه شده، در همان خانه آرایه ذخیره می شود.

۴ در قطعه کد مرحله ۳ تبدیل صریح به **int** را حذف کنید. چه خطایی رخ می دهد؟ چرا؟

برنامه ای بنویسید که هزینه روزانه یک ماه شما را دریافت کرده، در یک آرایه ذخیره کند. سپس هزینه ماهانه و میانگین هزینه روزانه شما را محاسبه کرده، نمایش دهد. سپس تعداد روزهایی که هزینه روزانه بیشتر از میانگین شده است را محاسبه کرده و نمایش دهد.

فعالیت منزل



## ارزشیابی مرحله ۲

مرحله کار	شرایط عملکرد (ابزار، مواد، تجهیزات، زمان، مکان و ...)	نتایج ممکن	استاندارد (شاخص ها/داوری/نمره دهی)	نمره
دسترسی به عناصر آرایه	مکان: کارگاه استاندارد رایانه تجهیزات: رایانه ای که نرم افزار IDE روی آن نصب است زمان: ۲۰ دقیقه	بالاتر از حد انتظار	نمایش عناصر آرایه - ویرایش عناصر آرایه - رفع خطای برنامه	۳
		در حد انتظار	نمایش عناصر آرایه - ویرایش عناصر آرایه	۲
		پایین تر از حد انتظار	نمایش عناصر آرایه	۱



## جست و جو در آرایه

فرض کنید در دفتر تلفن می‌خواهیم شماره تلفن یک شخص خاص را پیدا کنیم. اگر اسامی دفتر تلفن مرتب نباشد، لازم است از ابتدای دفتر تلفن نام مورد نظر را به ترتیب با تمام اسامی موجود در دفتر تلفن مقایسه کنیم. نام مورد نظر در هر کجای دفتر تلفن می‌تواند باشد، بنابراین عمل مقایسه تا یافتن نام مورد نظر و یا رسیدن به انتهای دفتر تلفن ادامه می‌یابد. به این روش **جست و جو، جست و جوی خطی یا ترتیبی (Linear Search)** می‌گویند. جست و جو یکی از عملیات رایج و پرکاربرد روی آرایه‌ها است. در جست و جو، شماره مکان عنصر مورد جست و جو مشخص می‌شود.

## پویانمایی شماره ۱۱۱۰۳: جست و جوی خطی

فیلم



فعالیت گروهی



یک آرایه نامرتب شامل اسامی ۱۰۰ هنرجو داریم. برای پیدا کردن نام یک هنرجو در این آرایه به روش جست و جوی خطی چند مقایسه نیاز است؟ جدول زیر را کامل کنید.

مکان عنصر مورد نظر	اول آرایه	وسط آرایه	آخر آرایه	در آرایه نباشد
تعداد مقایسه				

اگر اسامی دفتر تلفن بر اساس حروف الفبا مرتب باشند، به سرعت می‌توان نام مورد نظر را پیدا کرد. در این حالت وسط دفتر را باز می‌کنیم، اگر نام مورد نظر را دیدیم، جست و جو پایان یافته است، در غیر این صورت با توجه به ترتیب الفبایی و نام مورد نظر، در اسامی قبلی یا بعدی دفتر تلفن جست و جو را ادامه می‌دهیم. به این روش جست و جو، **جست و جوی دودویی (Binary Search)** می‌گویند.

فعالیت گروهی



در هر یک از موارد زیر تعیین کنید از روش جست و جوی خطی یا دودویی استفاده می‌شود؟ انتخاب روش براساس چه معیاری است؟

- ۱ یافتن کارنامه تحصیلی یک هنرجو در بین سایر کارنامه‌های مرتب شده براساس نام هنرجویان.
- ۲ پیدا کردن یک کتاب براساس نام آن کتاب در کتابخانه.
- ۳ پیدا کردن یک کتاب براساس کد کتاب در کتابخانه.
- ۴ پیدا کردن نام هنرجویی که در کلاس بالاترین معدل را دارد، اگر اسامی براساس معدل مرتب شده باشد.

می‌خواهیم برنامه‌ای برای نمایشگاه خودرو بنویسیم که پس از اطمینان از وجود یک خودرو در نمایشگاه، قیمت آن را در خروجی نمایش دهد.

۱ پروژه جدیدی به نام **LinearSearch** ایجاد کنید.

۲ برای خودروها و قیمت آنها دو آرایه متناظر اعلان کرده، مقداردهی کنید.

```
string[] car = {"Pride", "Mazda", "Renault", "Peugeot", "Toyota"};
```

```
long[] price = .....
```

۳ برای جست‌وجو در آرایه **car** کدهای زیر را بنویسید.

```
string item = "Pride";
for (int i = 0; i < car.Length; i++)
    if (item==car[i])
    {
        Console.WriteLine("{0} Found in {1}", item, i);
        break;
    }
```

متغیر **item** عنصر مورد جست‌وجو را نگه‌داری می‌کند.

برنامه را اجرا کنید. خروجی برنامه چیست؟

۴ برنامه را طوری تغییر دهید تا قیمت خودروی مورد جست‌وجو در خروجی نمایش داده شود.

۵ مقدار **item** را برابر **Hyundai** قرار دهید.

برنامه را اجرا کنید. خروجی چیست؟

۶ برای تشخیص یافتن و یا عدم یافتن عنصر، کد را به‌صورت زیر تغییر دهید.

```
string item = "Pride";
bool found = false;
for (int i = 0; i < car.Length; i++)
    if (item==car[i])
    {
        found = true;
        Console.WriteLine("price {0}={1}", item, price[i]);
        break;
    }
if (found == false)
    Console.WriteLine("item Not Found");
```

نقش متغیر **found** در این کد چیست؟

به‌جای شرط **found==false** عبارت **found**! را قرار دهید و برنامه را اجرا کنید.

۷ برنامه را طوری تغییر دهید که نام خودرو را از ورودی دریافت کند. به کمک کلید F10 برنامه را اجرا کنید. با دو مقدار Renault و Hyundai نتیجه برنامه را مشاهده کنید.

برنامه‌ای بنویسید که نام و نمره هنرجویان را از ورودی دریافت کرده، در آرایه مناسب ذخیره کند. سپس نام هنرجویانی را نمایش دهد که نمره ۲۰ گرفته‌اند.

فعالیت منزل



فیلم



فعالیت گروهی



پویانمایی شماره ۱۱۱۰۴: جست و جوی دودویی

### بازی حدس عدد

هنرجویی یک عدد بین ۱ تا ۱۰۰ انتخاب کرده، آن را یادداشت می‌کند. برنده شخصی است که با کمترین تعداد، عدد مورد نظر را حدس بزند.

- شما چه الگوریتمی برای برنده شدن پیشنهاد می‌دهید؟ بهتر است اولین حدس چه عددی باشد؟  
- هنرجویی عدد ۵۹ را یادداشت کرده است. در روش جست و جوی دودویی حداکثر با چند مقایسه عدد مورد نظر حدس زده خواهد شد؟ جدول زیر را کامل کنید.

مقایسه	نتیجه مقایسه	عدد حدس زده شده (وسط آرایه)	آخرین عدد آرایه	اولین عدد آرایه
۱	عدد بزرگ تر است	۵۰	۱۰۰	۱
۲				
۳				
۴				
۵				

- در جست و جوی دودویی اگر عنصر مورد جست و جو در آرایه نباشد، الگوریتم چگونه پایان می‌یابد؟

فعالیت منزل



با در نظر گرفتن عدد انتخابی ۱۲، جدول بالا را پر کنید.

کتابخانه غنی NET Framework دارای کلاس‌های آماده فراوانی است. این کلاس‌های آماده و از قبل ایجادشده (Built\_in) دارای اعضای مانند ویژگی‌ها و متدها هستند که انجام عملیات را آسان‌تر کرده، به توسعه سریع‌تر نرم‌افزار کمک می‌کنند. با کلاس Console و برخی متدهای آن مثل WriteLine و ReadLine آشنا هستیم. برای عملیات مختلف روی آرایه‌ها از کلاس آماده Array استفاده می‌کنیم.

### کارگاه ۷ متدهای کلاس Array

می‌خواهیم با یک مثال عملکرد متدهای کلاس Array جدول ۱ را بررسی کنیم.

جدول ۱- متدهای پرکاربرد کلاس Array

نام متد	کاربرد	شکل کلی
Sort		(نام آرایه) Array.Sort
Reverse		(نام آرایه) Array.Reverse
IndexOf		(مقدار، نام آرایه) Array.IndexOf
LastIndexOf		(مقدار، نام آرایه) Array.LastIndexOf
BinarySearch	یک مقدار را در آرایه صعودی مرتب‌شده جست‌وجو کرده، مکان اولین مورد پیدا شده را برمی‌گرداند. اگر پیدا نشود عدد منفی برمی‌گرداند.	(مقدار، نام آرایه) Array.BinarySearch

۱ یک پروژه جدید به نام ArrayClass ایجاد کنید.

۲ یک آرایه مفروض به‌صورت زیر اعلان و مقداردهی کنید.

```
int[] a = {20, 18, 10, 15, 50, 80, 10, 37};
```

۳ کدهای زیر را پس از اعلان وارد کنید.

```
int x = 10;
```

```
int found = Array.IndexOf(a, x);
```

```
Console.WriteLine(found);
```

مقدار متغیر found پس از اجرای برنامه چیست؟

عملکرد متد IndexOf چیست؟

۴ اگر  $x=60$  قرار دهیم خروجی چه می‌شود؟

۵ کد زیر را به برنامه اضافه کنید.

```
found = Array.LastIndexOf (a, x);
Console.WriteLine(found);
```

مقدار متغیر found پس از اجرای این دستور چیست؟

تفاوت عملکرد متد LastIndexOf با متد IndexOf چیست؟

۶ کد زیر را به برنامه اضافه کنید.

```
Array.Sort(a);
```

۷ با دستور foreach عناصر آرایه a را در خروجی نمایش دهید.

برنامه را اجرا کنید. عملکرد متد Sort چیست؟

۸ کد زیر را به برنامه اضافه کرده، برنامه را اجرا کنید.

```
x = 10;
found = Array.BinarySearch(a, x);
Console.WriteLine(found);
```

مقدار متغیر found چیست؟

۹ کد زیر را به برنامه اضافه کنید و دوباره با دستور foreach عناصر آرایه a را در خروجی نمایش دهید.

```
Array.Reverse(a);
```

برنامه را اجرا کنید. عملکرد متد Reverse چیست؟

۱۰ کد زیر را اضافه کرده، برنامه را اجرا کنید.

```
found=Array.BinarySearch(a, x);
Console.WriteLine(found);
```

مقدار متغیر found چیست؟

– می‌خواهیم برنامه‌ای بنویسیم که با استفاده از یک فهرست انتخاب (منو) عملیات زیر را روی آرایه انجام دهد و برنامه تا زمانی که کاربر گزینه خروج را انتخاب نکرده است، ادامه یابد. گزینه‌های منو به رنگ سفید روی زمینه آبی نمایش داده شوند.

فعالیت منزل



۱ دریافت عناصر آرایه Input

۳ مرتب کردن عناصر آرایه Sort

۵ جست‌وجوی خطی یک عنصر در آرایه Linear Search

۷ خروج Exit

۲ نمایش محتوای عناصر آرایه Show

۴ معکوس کردن عناصر آرایه Reverse

۶ جست‌وجوی دودویی یک عنصر در آرایه Binary Search



## فاز ۲:

برنامه کتابخانه هنرستان را با استفاده از دستوراتی که در این پودمان آموخته‌اید بنویسید.  
- تعداد کتاب‌ها و اعضا را از ورودی دریافت کرده، آرایه‌های متناظر برای آنها اعلان کنید.  
برای مشخصات اعضا، نام و نام خانوادگی و برای مشخصات کتاب، عنوان و موضوع کتاب را در نظر بگیرید و برای هر کدام یک آرایه مناسب طراحی کنید. اندیس آرایه‌ها به عنوان کد عضویت استفاده شود.  
- منوی زیر را طراحی کنید:

- ۱ Input user ( دریافت مشخصات اعضا )
- ۲ Input book ( دریافت مشخصات کتاب )
- ۳ User Show ( نمایش فهرست اعضا )
- ۴ Book Show ( نمایش فهرست کتاب‌ها )
- ۵ Book Subject ( نمایش فهرست کتاب‌ها با موضوع خاص )
- ۶ EditUser ( ویرایش مشخصات اعضا با دریافت کد عضویت آنها )
- ۷ Exit (خروج)

با اجرای برنامه این منو نمایش داده شود. کاربر برای انتخاب منوی مورد نظرش باید عدد متناظر منو را وارد کند. بعد از انجام عملیات منوی انتخاب شده، مجدداً منو نمایش داده شود.  
توجه: یک نوع داده شمارشی به نام MenuItem برای تشخیص منوی انتخاب شده تعریف کنید.  
- تکمیل پروژه: کد شماره ۱ و ۲ را در VS تایپ کرده و عملکرد آنها را مقایسه کنید.

شماره ۱	شماره ۲
<pre>string s = " C# Programing "; Console.Write(s);</pre>	<pre>string s = " C# Programing "; foreach (char c in s) {     Console.Write(c);     Thread.Sleep(50); }</pre> <p>برای استفاده از متد Sleep باید کد زیر را در بخش معرفی فضای نام بنویسید.</p> <pre>using System.Threading;</pre> <p>ورودی متد Sleep میزان توقف چاپ هر نویسه را بر حسب میلی ثانیه تعیین می‌کند. در این کد ۵۰ میلی ثانیه توقف دارد.</p>

برای زیباسازی پروژه از کدی مشابه کد شماره ۲ استفاده کنید.



برداشت



آنچه آموختم:

۱. ....
۲. ....
۳. ....

## ارزشیابی مرحله ۳



مراحل کار	شرایط عملکرد (ابزار، مواد، تجهیزات، زمان، مکان و ...)	نتایج ممکن	استاندارد (شاخص‌ها/داوری/نمره دهی)	نمره
جست‌وجو در آرایه	مکان: کارگاه استاندارد رایانه تجهیزات: رایانه‌ای که نرم‌افزار IDE برنامه نویسی روی آن نصب است زمان: ۳۰ دقیقه	بالاتر از حد انتظار	جست‌وجو در آرایه - مرتب‌سازی و معکوس کردن آرایه - رفع خطای برنامه	۳
		در حد انتظار	جست‌وجو در آرایه - مرتب‌سازی و معکوس کردن آرایه	۲
		پایین‌تر از حد انتظار	مرتب‌سازی و معکوس کردن آرایه	۱

**معیار شایستگی انجام کار:**

کسب حداقل نمره ۲ از مرحله اعلان آرایه

کسب حداقل نمره ۲ از بخش شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش

کسب حداقل میانگین ۲ از مراحل کار

## جدول ارزشیابی پایانی

<p><b>شرح کار:</b></p> <p>۱ اعلان آرایه</p> <p>۲ دسترسی به عناصر آرایه</p> <p>۳ جست و جو در آرایه</p>																											
<p><b>استاندارد عملکرد:</b></p> <p>با استفاده از دانش ساختار آرایه مسئله را تحلیل کرده، برنامه را ایجاد و خطایابی کند.</p>																											
<p><b>شاخص‌ها:</b></p> <table border="1"> <thead> <tr> <th>شماره مرحله کار</th> <th>شاخص‌های مرحله کار</th> </tr> </thead> <tbody> <tr> <td>۱</td> <td>اعلان آرایه - رفع خطای برنامه</td> </tr> <tr> <td>۲</td> <td>به کارگیری عناصر آرایه - رفع خطای برنامه</td> </tr> <tr> <td>۳</td> <td>به کارگیری متدهای مرتب‌سازی آرایه - جست و جو در آرایه - رفع خطای برنامه</td> </tr> </tbody> </table>				شماره مرحله کار	شاخص‌های مرحله کار	۱	اعلان آرایه - رفع خطای برنامه	۲	به کارگیری عناصر آرایه - رفع خطای برنامه	۳	به کارگیری متدهای مرتب‌سازی آرایه - جست و جو در آرایه - رفع خطای برنامه																
شماره مرحله کار	شاخص‌های مرحله کار																										
۱	اعلان آرایه - رفع خطای برنامه																										
۲	به کارگیری عناصر آرایه - رفع خطای برنامه																										
۳	به کارگیری متدهای مرتب‌سازی آرایه - جست و جو در آرایه - رفع خطای برنامه																										
<p><b>شرایط انجام کار و ابزار و تجهیزات:</b></p> <p><b>مکان:</b> کارگاه رایانه مطابق استاندارد تجهیزات هنرستان‌ها</p> <p><b>تجهیزات:</b> رایانه‌ای که نرم‌افزار IDE برنامه‌نویسی روی آن نصب است.</p> <p><b>زمان:</b> ۶۰ دقیقه (اعلان آرایه ۱۰ دقیقه - دسترسی به عناصر آرایه ۲۰ دقیقه - جست و جو در آرایه ۳۰ دقیقه)</p>																											
<p><b>معیار شایستگی:</b></p> <table border="1"> <thead> <tr> <th>ردیف</th> <th>مرحله کار</th> <th>حداقل نمره قبولی از ۳</th> <th>نمره هنجار</th> </tr> </thead> <tbody> <tr> <td>۱</td> <td>اعلان آرایه</td> <td>۲</td> <td></td> </tr> <tr> <td>۲</td> <td>دسترسی به عناصر آرایه</td> <td>۱</td> <td></td> </tr> <tr> <td>۳</td> <td>جست و جو در آرایه</td> <td>۱</td> <td></td> </tr> <tr> <td colspan="2"> <p>شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش:</p> <p>تصمیم‌گیری، تشخیص اهداف و محدودیت‌ها - زبان فنی</p> <p>رعایت ارگونومی</p> <p>حفاظت از تجهیزات کارگاه</p> <p>دقت در تشخیص ساختار مناسب داده هنگام تعریف آرایه در برنامه</p> </td> <td>۲</td> <td></td> </tr> <tr> <td colspan="3"> <p><b>میانگین نمرات</b></p> </td> <td>*</td> </tr> </tbody> </table>				ردیف	مرحله کار	حداقل نمره قبولی از ۳	نمره هنجار	۱	اعلان آرایه	۲		۲	دسترسی به عناصر آرایه	۱		۳	جست و جو در آرایه	۱		<p>شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش:</p> <p>تصمیم‌گیری، تشخیص اهداف و محدودیت‌ها - زبان فنی</p> <p>رعایت ارگونومی</p> <p>حفاظت از تجهیزات کارگاه</p> <p>دقت در تشخیص ساختار مناسب داده هنگام تعریف آرایه در برنامه</p>		۲		<p><b>میانگین نمرات</b></p>			*
ردیف	مرحله کار	حداقل نمره قبولی از ۳	نمره هنجار																								
۱	اعلان آرایه	۲																									
۲	دسترسی به عناصر آرایه	۱																									
۳	جست و جو در آرایه	۱																									
<p>شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش:</p> <p>تصمیم‌گیری، تشخیص اهداف و محدودیت‌ها - زبان فنی</p> <p>رعایت ارگونومی</p> <p>حفاظت از تجهیزات کارگاه</p> <p>دقت در تشخیص ساختار مناسب داده هنگام تعریف آرایه در برنامه</p>		۲																									
<p><b>میانگین نمرات</b></p>			*																								

\* حداقل میانگین نمرات هنجار برای قبولی و کسب شایستگی، ۲ است.