

پودمان ۲

واحد یادگیری ۳ و ۴

مدیریت مجموعه داده

تدریس واحد یادگیری ۳

شایستگی کار با ساختار تکراری

مقدمات تدریس

الف) مفاهیم کلیدی

مفاهیم کلیدی				
شمارنده حلقه	Trace	خروج از حلقه	حلقه‌های متداخل	حلقه
حلقه نامعین	خروجی‌های حلقه	حلقه معین	ورودی‌های حلقه	Break

ب) مراحل انجام کار واحد یادگیری ۳

مراحل انجام کار	
۱	حل مسئله تکرار
۲	ایجاد برنامه با حلقه‌های for
۳	خطایابی با استفاده از ابزارها
۴	ایجاد برنامه با حلقه‌های while
۵	ایجاد برنامه با حلقه‌های متداخل

ج) تجهیزات لازم

مشخصات سخت‌افزاری برای نصب نرم‌افزار Visual Studio Express 2012:

- پردازنده حداقل ۱/۶ گیگاهرتز
- حافظه RAM حداقل ۱ گیگابایت
- حداقل فضای موجود در دیسک سخت ۴ گیگابایت
- کارت ویدئویی متناسب با DirectX 9 و قابل اجرا در رزولوشن 1024×768 و بالاتر
- مطمئن شوید که نرم‌افزار VS در تمام رایانه‌های موجود در کارگاه فعال (Active) است.

د) بودجه‌بندی

برای بودجه‌بندی فصل دوم می‌توانید از نمونه پیشنهادی استفاده کنید.

جلسه	واحد یادگیری	کارگاه (موضوع)	شماره صفحات	اهداف توانمندسازی	فعالیت‌های تکمیلی
۸	۳	کارگاه ۳-۲-۱	۶۵-۶۷	آشنایی با حلقه معین و استفاده از آن برای حل مسئله - بررسی شرایط حلقه معین، تغییر در مقادیر ابتدایی و انتهای و شرط اجرای حلقه - ایجاد حلقه‌های تأخیر در مسئله با استفاده از حلقه معین	از هم‌کلاسی خودتان بخواهید سه بار اسم خودش را بنویسد.
۹	۳	کارگاه ۵-۴	۶۷-۷۰	کار با متغیر خارج بلاک - کار با دستورات داخل حلقه - ایجاد خروجی‌های مناسب در داخل و بیرون حلقه - کار با دستور break - ایجاد وقفه در روند اجرای حلقه تکرار	برنامه محاسبه مجموع تعداد خودکارهای قرمز و آبی دانش‌آموزان یک کلاس.
۱۰	۳	کارگاه ۷-۶	۷۴-۷۶	تولید خروجی‌های خاص مانند اعداد زوج و فرد - کار با حلقه‌های نامعین - تبدیل حلقه‌های معین به نامعین - جستجوی اعداد خاص	برنامه محاسبه تعداد دانش‌آموزان با رایانه‌های شماره فرد یا زوج
۱۱	۳	کارگاه ۹-۸	۷۷-۷۹	کار با حلقه‌های متداخل - توسعه حلقه‌های متداخل - آشنایی با مفاهیم حلقه‌های متداخل - استفاده از این نوع حلقه‌ها در حل مسائل روزمره	برنامه جدول ضرب اعداد - برنامه نمایش یک ساعت دیجیتال - برنامه خودپرداز
	۴	کارگاه ۲و۱	۸۵-۸۷	آشنایی با مفهوم و کاربرد آرایه - شناخت انواع آرایه - تعریف یک آرایه - شناخت عناصر آرایه - مقداردهی به عناصر آرایه	برنامه‌ای حروف نام خود را به صورت تک‌تک در یک جدول بنویسد و تعیین حرف چهارم اسم شما؟
۱۲	۴	کارگاه ۵و۴و۳	۸۹-۹۱	چگونگی دسترسی به مقادیر عناصر آرایه - دسترسی به یک عنصر خاص در آرایه - نمایش مقادیر ذخیره‌شده در آرایه - نمایش محتوای یک عنصر آرایه -	برنامه‌ای که حرف سوم اسم شما را بنویسد. برنامه‌ای بنویسد که رقم‌های یک عدد

فعالیت‌های تکمیلی	اهداف توانمندسازی	شماره صفحات	کارگاه (موضوع)	واحد یادگیری	جلسه
چهار رقمی را تک تک ذخیره کند.					
برنامه‌ای که با استفاده از روش جستجوی خطی نام دانش‌آموزی که شغل پدرش معلم (Teacher) است را بیابد - برنامه‌ای که با استفاده از مفهوم Flag وجود یا عدم وجود حرف a را در نام یک دانش‌آموز را بررسی کنید.	جستجوی مقدار یا مقادیر خاصی در خانه‌های آرایه - ویرایش محتوای خانه‌های آرایه - شناخت روش جستجوی خطی - شناخت روش جستجوی دودویی - شناخت مکانیسم Flag - کد نویسی یکی از روش‌های جستجوی خطی یا دودویی	۹۱-۹۵	کارگاه ۷و۶	۴	۱۳
برنامه‌ای که با استفاده از متدهای کلاس آرایه مقادیر ذخیره‌شده در آرایه را به صورت صعودی مرتب می‌کند و مقدار دلخواه کاربر را در آن جستجو می‌کند. برنامه‌ای که با استفاده از داده‌های شمارشی عدد مربوط به ماه وارد شده از طرف کاربر را نمایش می‌دهد .	شناخت کلاس آرایه - شناخت متدهای کلاس آرایه - بکارگیری متدهای کلاس آرایه در برنامه - تعریف داده شمارشی - معرفی یک داده شمارشی در برنامه - استفاده از داده شمارشی در برنامه	۹۶-۱۰۳	کارگاه ۹و۸و۱۰و۱۱	۴	۱۴

زمان‌بندی واحدهای یادگیری:

زمان لازم برای تدریس هر بخش از واحد یادگیری ۳ در جدول زیر پیشنهاد می‌شود.

ردیف	اهداف توانمندسازی	سهم زمان
۱	مفهوم حلقه را درک کند.	۵ ساعت
۲	انواع حلقه را نام ببرد.	
۳	بتواند روند نمای یک کد حلقه را ایجاد کند.	
۴	بتواند روند نما را به حلقه تبدیل کند.	۱۰ ساعت
۵	ساخت و پیاده سازی حلقه‌های معین حلقه معین را تشخیص داده، روند نمای آن را ایجاد و به الگوریتم تبدیل کند - کد نویسی حلقه‌های معین را انجام داده و اشکال‌زدایی کند.	
۶	ساخت و پیاده‌سازی حلقه‌های نامعین حلقه نامعین را تشخیص داده، روند نمای آن را ایجاد و به الگوریتم تبدیل کند - کد نویسی حلقه‌های نامعین را انجام داده و اشکال‌زدایی کند.	۱۰ ساعت
۷	بکارگیری حلقه‌های متداخل و دستورات کنترلی حلقه بتواند از حلقه‌های متداخل استفاده کند- برنامه را به صورت خطی اجرا و اشکال‌زدایی کند - شرایط خروج از حلقه را درک کرده و از دستورات آن استفاده کند.	۵ ساعت
	مجموع زمان	۳۰ ساعت

طرح درس روزانه:

نمونه طرح درس پیشنهادی واحد یادگیری سوم :

طرح درس روزانه (هفتگی) پیشنهادی				
درس: ساختارهای تکرار		کلاس: یازدهم		
پیام جلسه (هدف کلی): هنرجو بتواند مسئله تکرار را تشخیص داده و آن را با حلقه حل کند .				
زمان فیزیکی	فعالیت‌ها		اهداف یادگیری	
مدت (دقیقه)	کار هنرجو	کار هنرآموز	طبقه هدف: حیطه عاطفی / شناختی / روانی - حرکتی	فعالیت
۳۰	مشارکت در پاسخگویی و تعامل	نشان دادن انواع حلقه و مسائل حل شده با حلقه	سنجش میزان آگاهی هنرجویان از کد نویسی و آشنایی با مفهوم حلقه‌ها	ارزشیابی رفتار ورودی
۳۰	یک‌روند نمای حلقه ترسیم کند .	با استفاده از حلقه یک مسئله را حل کند .	نمایش فیلم	ایجاد انگیزه
۸۰	هنرجویان مطالب پایه را می‌آموزند.	مفهوم حلقه‌ها را کامل توضیح داده و انواع حلقه‌ها را نام برده و مثال مناسب بنویسد تفاوت حلقه‌ها را بیان کرده و نکات مربوط به حلقه‌ها را توضیح دهد .	توضیح کامل مفاهیم کلیدی (دانشی) و ایجاد علاقه و انگیزه در هنرجویان (بینشی)	ارائه مفاهیم کلیدی (توضیح هنرآموز)
۵۰	با توجه به فعالیت‌های کارگاهی کتاب هنرجویان در گروه‌های خود، گزینه‌های دیگر را با آزمون‌وخطا می‌آموزند.	هنرآموز توضیح کلی در مورد فعالیت کارگاهی به هنرجو داده و از آن‌ها می‌خواهد به صورت گروهی به انجام آن بپردازند.	هنرجو باید توانایی کار با حلقه‌ها را داشته باشد و فعالیت‌های کارگاهی و گروهی را انجام دهد .	فعالیت کارگاهی (تمرین هنرجویان)
۳۰	هنرجویان در گروه‌های خود سعی می‌کنند با نهایت دقت و سرعت با ایجاد خلاقیت به انجام فعالیت کارگاهی بپردازند.	از هنرجویان خواسته می‌شود در گروه‌های تعیین شده به انجام فعالیت‌های کارگاهی بپردازند. قبل از شروع به کار هنرجویان، ملاک‌های ارزیابی تمرین‌ها به آن‌ها داده شود.	طرح یک مسئله که با استفاده از حلقه حل می‌شود .	ارزیابی فعالیت‌ها ارائه تمرین

طرح درس روزانه (هفتگی) پیشنهادی			
کلاس: یازدهم		درس: ساختارهای تکرار	
پیام جلسه (هدف کلی): هنرجو بتواند مسئله تکرار را تشخیص داده و آن را با حلقه حل کند.			
اهداف یادگیری	فعالیت‌ها	زمان فیزیکی	
ارائه نکات تکمیلی (جمع‌بندی)	هنرجو باید مسئله را شناخته و حلقه مناسب را انتخاب کرده و روند نمای مربوط به آن را ترسیم و سپس کد نویسی کند.	۴۰	توجه، دقت، پرسش و پاسخ- انجام فعالیت‌های گروهی و ارائه راه‌حل‌های جدید برای مسائل.
ارزشیابی شایستگی (ارزشیابی پایانی)	هنرجو باید بتواند یک مسئله را شناخته و حلقه مناسب برای حل آن انتخاب کرده، کد نویسی را عملی انجام دهد	۴۰	این ارزشیابی در دوشاخه انفرادی و گروهی انجام می‌شود: از هنرجویان خواسته می‌شود در گروه‌های تعیین شده به انجام پروژه این بخش از واحد کار بپردازند. نمره‌ای به کار گروهی هنرجویان داده می‌شود. آزمون برای هنرجویان به صورت انفرادی برگزار می‌شود. نمره گروهی در نمره انفرادی افراد تأثیرگذار است. نکته: قبل از شروع به کار، ملاک‌های ارزیابی پروژه به آن‌ها داده می‌شود.
ابزارهای موردنیاز	ویدئو پروژکتور، رایانه، تخته آموزشی، برنامه VS		

ه) فیلم‌ها و پرونده‌های پیوست

شماره فیلم	هدف
111h7	کار حلقه WHILE
111h8	کار حلقه FOR
111h9	کار با BREAK
111h10	برنامه ماکزیمم
111h11	کار با حلقه‌های متداخل
111h12	تعریف آرایه
111h13	کاربرد آرایه
111h14	ویژگی length در آرایه
111h15	مرتب‌سازی آرایه
111h16	جستجوی خطی
111h17	جستجوی دودویی
111h18	کلاس array

نکته: برای انجام فعالیت‌های تکمیلی می‌توانید از کدهای نوشته‌شده موجود در «پرونده پیوست فعالیت‌های تکمیلی» استفاده کنید.

و) ورود به بحث

قبل از تدریس این واحد یادگیری بهتر است هنرجو را با مفاهیم «تکرار» و «حلقه» آشنا کنید.

از دیرباز انجام فعالیت‌های مشابه خسته‌کننده و وقت‌گیر بوده است. تکرار بخش جدانشدنی برنامه‌نویسی است که به برنامه‌نویسان کمک خواهد کرد تا از انجام فعالیت‌های مشابه پرهیز کنند. در اطراف و زندگی روزمره ما کارهای تکراری بسیاری به چشم می‌خورد. کارمندی که هرروز صبح به محل کار خود می‌رود، هنرجویانی که هر هفته در مدرسه حاضر می‌گردند، روزهای هفته، روزهای سال و ... از تکرارهای طبیعی زندگی ما انسان‌ها است. به هنرجوی خود یاد دهید که در روند یادگیری این واحد مثال‌هایی از زندگی عادی و روزمره که دارای تکرار به هر شکلی می‌باشند را یادداشت‌برداری کرده و در کلاس درس ارائه کند.

ایجاد انگیزه در هنرجویان :

از هنرجویان خود بخواهید که پنج بار نام خودشان را بر روی کاغذ یادداشت کنند. حال از آن‌ها بخواهید مراحل انجام این فعالیت را به صورت یک الگوریتم بنویسند. سپس روندنمای این مراحل را ترسیم کنند.

برای تشریح بهتر مفهوم واحد یادگیری به گروه اول بگویید این کار را ۱۰ بار و به گروه دوم بگویید این کار را ۲۰ بار و همین‌طور الی آخر انجام دهند. از آن‌ها بخواهید بنویسند هر گروه چند بار کار تکراری کرده‌اند؟ به هر گروه چند خط الگوریتم و چند شکل اضافه شده است؟ به منظور تعیین سطح دانش آموزان می‌توانید سؤالاتی از این دست مطرح کنید.

سؤال ۱: به نظر شما در محیط مدرسه چه کارهای تکراری انجام می‌شود؟

سؤال ۲: کدام یک از شما می‌تواند یک مثال برای کار تکراری بزند؟ سپس بخواهید تا مراحل تکرار را مشخص کنند و تعداد آن‌ها را بشمارند. از آن‌ها سؤال کنید آیا ابتدا و انتهای این کارها مشخص است یا نامشخص؟

سؤال ۳: چرا به تکرار در کارهایمان احتیاج داریم؟

سؤال ۴: آیا تابه حال در صف نانواپی ایستاده‌اید؟

سؤال ۵: آیا تابه حال از جدول ضرب استفاده کرده‌اید؟

سؤال ۶: آیا می‌توانید اعداد ۱۰۰ تا ۳۰۰ را سه تا سه تا بشمارید.

سؤال ۷: فرض کنید در یک کارخانه سازنده ماشین برای هر ماشین چهار حلقه تاپر گذاشته می‌شود. آیا این یک کار تکراری است؟

سؤال ۸: از شما خواسته می‌شود تا صفحاتی از یک کتاب را بخوانید. شما ممکن است اصلاً صفحه‌ای را مطالعه نکنید یا به تعداد دلخواه، صفحاتی از کتاب را مطالعه کنید.

سؤال ۹: آیا تابه حال با دوستانتان بازی گل‌یاپوچ را انجام داده‌اید؟ آیا به جدول لیگ برتر فوتبال ایران نگاه کرده‌اید؟ تعداد ست‌های یک مسابقه والیبال معین است یا نامعین؟

سؤال ۱۰: به ساعت کارگاه نگاه کنید و تکرارها را تشخیص دهید.

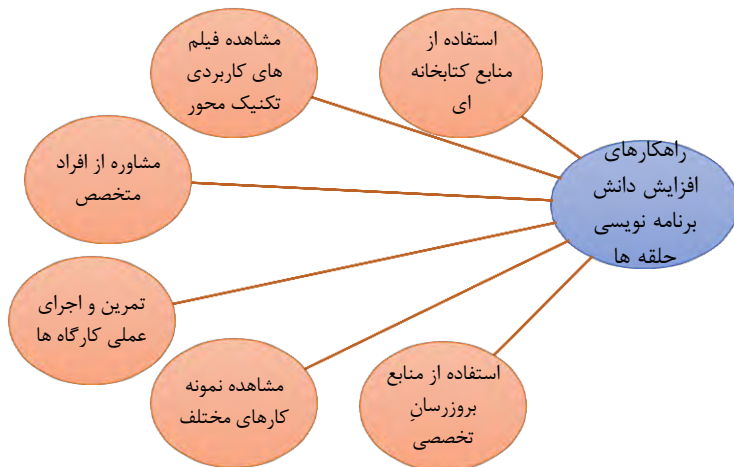
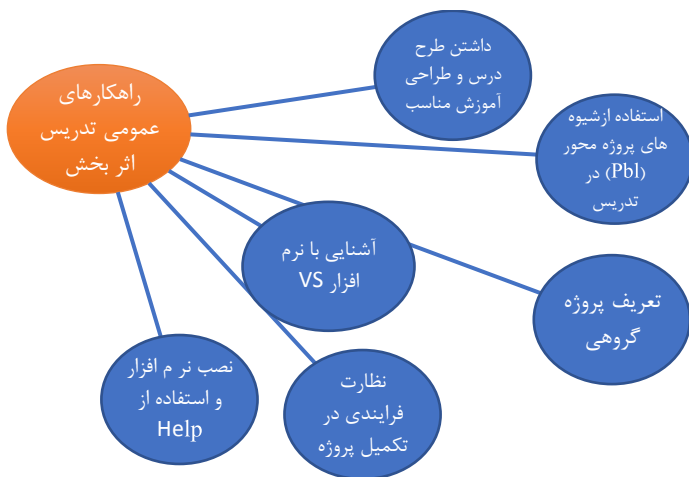
سؤال ۱۱: چند نمونه از کاربردهای تکرار را نام ببرید؟

سؤال ۱۲: ۱۰۰ بار واژه سلام را بر روی دفتر بنویسید.

سؤال ۱۳: ۱۰ نماد روی صفحه رومیزی ویندوز خود را در یک گوشه جمع کنید.

سؤال ۱۴: ۱۰۰ نقطه برای بازی نقطه - خط بر روی کاغذ خود بگذارید.

هنگامی که با کارهای مشابه برخورد می‌کنید به این فکر خواهید کرد که چطور می‌شود کار را یک‌بار انجام داد سپس به تعداد مناسب آن را تکرار کرد. تکرارهای طبیعی زندگی به صورت تکرارهای مشخص و یا غیرقابل پیش‌بینی رخ می‌دهند.

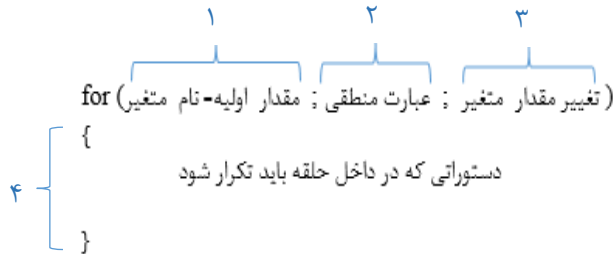


تدریس

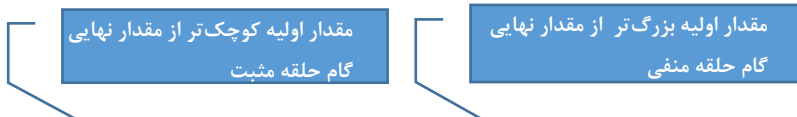
حلقه ها

حلقه تعداد تکرار یک کار است که در دو نوع معین و نامعین تعریف شده است. وقتی سه بار سلام می‌کنید از یک حلقه استفاده کرده‌اید. یا زمانی که منتظر نتیجه یک مسابقه والیبال هستید یک حلقه نامعین را دنبال کرده‌اید. وقتی به ناوایی

می‌روید و در صف قرار می‌گیرید که نوبت شما رسیده و نان را سفارش دهید، در واقع از دو نمونه حلقه نامعین و معین استفاده کرده‌اید. هر حلقه for دارای چهار بخش است. (متغیر و مقداردهی اولیه حلقه، عبارت منطقی، مقدار گام حلقه، دستورات تکرارشونده) شکل دستوری برنامه‌نویسی حلقه for به صورت زیر است.



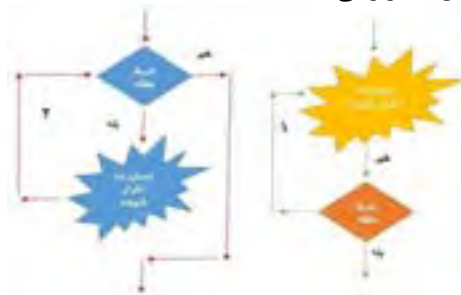
در حلقه for می‌توان از اعداد صحیح و اعشاری، کاراکتر و عبارت در قسمت‌های مقدار اولیه، گام حلقه و مقدار نهایی استفاده کرد. حلقه‌های معین را می‌توانیم به دو صورت افزایشی یا کاهشی بنویسیم.



حلقه افزایشی

حلقه کاهشی

حلقه نامعین حلقه‌ای است که تعداد تکرار آن معین و مشخص نیست. به‌طور کلی دو نمونه حلقه نامعین را در شکل‌های زیر می‌بینید.



شکل شماره ۱:

حلقه‌ای است که ابتدا دستورات تکرارشونده داخل حلقه انجام می‌شوند و سپس شرط حلقه بررسی می‌شود. تعداد تکرار این حلقه، یک‌بار یا بیشتر از یک‌بار است. مثال: برنامه‌ای بنویسید تا زمانی که عدد ۱۰۰ را نگفته‌اید از شما عدد پرسیده شود:

```
byte n;  
do  
{  
n =  
byte.Parse(Console.ReadLine());  
} while (n !=100);
```

شکل شماره ۲:

حلقه‌ای است که ابتدا شرط حلقه بررسی شده و سپس در صورتی که جواب شرط درست باشد دستورات داخل حلقه اجرا می‌شوند و در صورتی که شرط درست نباشد حلقه به پایان می‌رسد. تعداد تکرار این حلقه، صفر بار یا بیشتر است.

مشکلات متداول در فرآیند یاددهی - یادگیری

این واحد یادگیری به دلیل تأثیر واضح یا پنهان در سایر واحدهای کتاب از اهمیت فوق‌العاده‌ای برخوردار است. شاید بزرگ‌ترین دغدغه هنرجویان در فراگیری این درس این باشد که کجا از حلقه معین و کجا از حلقه نامعین استفاده کنند؟ درست است که تشخیص استفاده مناسب از حلقه‌ها با تمرین به دست می‌آید ولی نشانی‌هایی مانند میان، بار، مرتبه، تا، بین و از این دست کلمات در برنامه‌ها هنرجو را به استفاده از حلقه معین و کلماتی شبیه تا زمانی که، تا وقتی که، تا هنگامی که و ... که معنای انتظار نامشخص را می‌دهند ایشان را به استفاده از حلقه نامعین هدایت می‌کنند.

شیوه و الگوی پیشنهادی:

برنامه‌نویسی بدون داشتن شیوه و الگوی خاص تقریباً غیرممکن است. سعی کنید پیش‌زمینه‌های مناسب را برای فراگرفتن این واحد یادگیری در هنرجو ایجاد کنید. پیشنهاد می‌شود به هنرجویان تکالیف گروهی و البته مختلف داده شود و همچنین ذهن آن‌ها را با مسئله‌های تکرار روزمره درگیر کنید. هم چنین پیشنهاد می‌شود ابتدا به هنرجویان روش طراحی و پیاده‌سازی بر روی کاغذ را آموزش داده و سپس اجازه دسترسی به سیستم داده شود.

پاسخ به فعالیت‌ها :

فعالیت کارگاهی
ص ۶۱

در جدول زیر مشخص کنید کدام یک از آن‌ها معین و نامعین هستند.

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

مثال
ص ۶۲

یک هنرآموز، فعالیت منزل هنرجویان را تحویل گرفته، پس از بررسی گزارش هر کدام را در دفتر ارزشیابی خود وارد می‌کند.

ورودی‌ها	فعالیت‌های انجام شده هنرجویان	شناخت مسئله
خروجی‌ها	ثبت گزارش هر هنرجو در دفتر ارزشیابی	
بدنه حلقه	دریافت فعالیت، بررسی فعالیت، ثبت گزارش برای هنرجو، تحویل فعالیت به هنرجو	

مسئله
ص ۶۳

می‌خواهیم سیستم نوبت دهی یک مطب دندان پزشکی را مکانیزه کنیم، به طوری که برای نوبت دهی به ۲۰ بیمار خود از شماره‌های چاپ شده ۱ تا ۲۰ استفاده کند.

پاسخ :

۱. شروع

۲. مقدار i را برابر یک قرار بده

۳. اگر $i \leq 20$ برو به مرحله بعد وگرنه برو به مرحله ۶

۴. مقدار i را نمایش بده.

۵. i را یک واحد اضافه کن

۶. برو به مرحله ۲

۷. پایان

فعالیت کارگاهی
ص ۶۳

- اگر تعداد بیماران ۳۰ نفر باشد، چه تغییری لازم است؟
خط دوم: ۲- اگر $i \leq 30$ برو به مرحله بعد وگرنه برو به مرحله ۶

- روندنما و الگوریتم را تغییر دهید تا تعداد بیماران را از ورودی دریافت کند.
شروع

مقدار n را دریافت کن.

مقدار i را برابر یک قرار بده

.... اگر $i \leq n$ برو به مرحله ۴ وگرنه برو به مرحله ۷
 مقدار i را نمایش بده.
 i را یک واحد اضافه کن
 برو به مرحله ۳
 پایان

کارگاه ۱ – تبدیل روند نما به برنامه

پاسخ به فعالیت‌ها

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

تکمیل کارگاه
ص ۶۵

پاسخ:

```
int i;
for (i = 100; i <= 999; i++)
    Console.WriteLine(i);
```

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

پاسخ:

```
int i;
int n;
n = int.Parse(Console.ReadLine());
for ( i = 1 ; i <= n ; i++)
    Console.WriteLine(i);
```

کارگاه ۲ – بررسی شرایط حلقه تکرار معین

پاسخ به فعالیت‌ها

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

تکمیل کارگاه
ص ۶۶

پاسخ:

در صورتی که در انتهای دستور حلقه معین for از علامت ; استفاده کنیم، بدنه حلقه اجرا نخواهد شد ولی متغیر افزایش یا کاهش خواهد یافت. در واقع با این کار شما یک حلقه تأخیری افزایشی یا کاهش‌ی ایجاد کرده‌اید.
 - بررسی کنید در صورتی که آکولادهای قطعه کد بالا حذف شوند خروجی چه تغییری می‌کند؟

پاسخ:

فقط یک دستور و آن هم نزدیک‌ترین دستور به for اجرا خواهد شد.

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

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

در این قطعه کد حلقه گردش خواهد کرد و در انتهای گردش حلقه دستور نمایش متغیر i ، آخرین مقدار i یعنی مقدار ۳۲ نمایش داده خواهد شد.
- قطعه کد مرحله ۲ را طوری تغییر دهید تا شماره هنرجویانی که باید کارگاه را نظافت کنند چاپ کند.

```
Byte i;
```

```
for (i = 1; i <= 30; i += 2)
```

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

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

پاسخ: برنامه خطای منطقی دارد و خروجی نمایش داده نمی‌شود، زیرا شرط قسمت دوم حلقه برقرار نیست.

کنجکاوی
ص ۶۶

آیا در VS با دو بار فشردن کلید Tab پس از دستور if و switch، شکل کامل دستور نوشته می‌شود؟
پاسخ: بله، با فشردن دو بار کلید Tab شکل کامل دستورات اضافه خواهد شد.

فعالیت کارگاهی
ص ۶۶

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

خروجی برنامه پس از اجرا	خروجی برنامه از نظر شما	حلقه
چاپ ۵۴۳۲۱		<pre>int i; for (i=5;i>=1;i--) Console.Write(i);</pre>
چاپ ۰-۳-۶-۹ زیر هم		<pre>int i; for (i=0;i<10;i+=3) Console.WriteLine(i);</pre>
چاپ اعداد زوج نزولی از ۱۰ تا ۰ به صورت زیر هم		<pre>int i; for (i=10;i>=0;i-=2) Console.WriteLine(i)</pre>

جدول زیر را کامل کنید.

حلقه	نتیجه اجرا
<pre>int i; for (i = 100; i <= 1000; i += 100) Console.WriteLine("ali");</pre>	
<pre>int i; int s=0; for (i = 100; i <= 1000; i += 100) s = s + i; Console.WriteLine("sum{0}", s);</pre>	
<pre>byte i; for (i = 100; i >= 10 ; i -= 10) Console.WriteLine(i);</pre>	
<pre>char ch ; for (ch = 'b'; ch <= 'k'; ++ch) Console.WriteLine(ch);</pre>	

آیا متغیر حلقه می‌تواند از نوع float یا char باشد؟
پاسخ:
بله، به این مثال‌ها توجه کنید.

```
float i;
for ( i = 1.5f ; i <= 3.75 ; i=i+0.25f)
    Console.WriteLine(i);
char i;
for ( i = 'a' ; i <= 'g' ; i++)
    Console.WriteLine(i);
```

هنرآموزی می‌خواهد ده نفر از هنرجویان یک کلاس را به گروه‌های 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. برنامه‌ای بنویسید که شماره هنرجویان هر گروه را در خروجی نمایش دهد.
پاسخ:

```
byte i;
byte j=10;
for (i = 1; i <= 5; i++)
{
    Console.WriteLine("{0} --> {1}",i,j);
    j--;
}
```

کارگاه ۳ - کار با متغیر خارج بلاک

پاسخ به فعالیت‌ها

سوال
ص ۶۷

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.
پاسخ: چون متغیر `i` در حلقه تعریف و مقداردهی شده ولی در خارج از حلقه استفاده شده باعث بروز خطا شده است. برای حل این مشکل تعریف متغیر `i` را در ابتدای برنامه انجام داده، برنامه را بدون خطا اجرا کنید.

فعالیت‌های
تکمیلی

مسئله شماره ۱: برنامه‌ای بنویسید که ۱۰ عدد را بخواند و با وارد شدن اولین عدد مضرب ۵ برنامه پایان بپذیرد.
مسئله شماره ۲: برنامه‌ای بنویسید که دونده‌ای بتواند ۱۰ دور حیاط مدرسه را بدود اما در دور ششم بایستد و ادامه ندهد.

کارگاه ۴ - محاسبه مجموع

پاسخ به فعالیت‌ها

تکمیل کارگاه
ص ۶۸

- برنامه را اجرا کنید. چه مقداری در خروجی نمایش داده می‌شود؟
پاسخ: ۱۰
- برنامه را به صورت زیر تغییر دهید و برنامه را اجرا کنید. آیا در خروجی تغییری صورت گرفته است؟

```
NumberOfGame : 0
NumberOfGame : 4
NumberOfGame : 7
NumberOfGame : 9
```

- برنامه را طوری تغییر دهید که تعداد کل بازی‌های لیگ برتر را نمایش دهد.

```
for (int i = 15; i >= 1; i--)
```

- برنامه را طوری تغییر دهید که تعداد گروه‌ها را از ورودی دریافت کند.

```
byte n;
int numberOfGame = 0;
n = byte.Parse(Console.ReadLine());
for (int i = n-1; i >= 1; i--)
    numberOfGame += i;
Console.WriteLine("NumberOfGame: " + numberOfGame);
```

کارگاه ۵ - خروج زودرس از حلقه

در مواقعی که نیاز دارید برنامه زودتر از موعد از بلاک خارج شود می‌توانید از دستور `break` برای خروج زودرس از حلقه استفاده کنید.

```
for (int i=10 ; i<=100 ; ++i)
if (i%10==0) break;
```

پاسخ به فعالیت‌ها

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

تکمیل کارگاه

ص ۶۹

```
int guess;
int month;
int score = 5;
Console.WriteLine("Enter number your month:");
month = byte.Parse(Console.ReadLine());
Console.Clear();
for (byte i = 1; i <= 5; i++)
{
    Console.WriteLine("your guess? ");
    guess = byte.Parse(Console.ReadLine());
    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 ! ");
    }
}
```

اگر مقدار month را بازده وارد کنیم (month = 11)

i	guess	score	خروجی
1	3	4	Try again !

2	8	3	Try again !
3	2	2	Try again !
4	4	1	Try again !
5	5	0	Try again!

در صورتی که مقدار ورودی برای متغیر month رقم ۵ باشد خروجی به ازای ورودی‌های جدول داده‌شده به شکل زیر است.

```
your guess? 3
try again !
your guess? 8
try again !
your guess? 5
you win
your score: 3
```

- برنامه را طوری تغییر دهید که پس از ۵ بار حدس اشتباه کاربر دوم، پیام مناسب دیگری چاپ‌شده، شماره ماه تولد را نمایش دهد.
پاسخ:

```
Console.Clear();
for (byte i = 1; i <= 5; i++)
{
    Console.Write("your guess? ");
    guess = byte.Parse(Console.ReadLine());
    if (guess == month)
    {
        Console.ForegroundColor =
ConsoleColor.Green;
        Console.WriteLine("you win ");
        Console.WriteLine("your score: {0} ",
score);
        break;
    }
    else
    {
        score -= 1;
        if (score == 0)
        {
            Console.WriteLine("you lose");
            Console.WriteLine(" month is: " + month);
```

```

        break;
    }
    Console.WriteLine("try again ! ");
}
}

```

- برنامه مرحله ۵ را طوری تغییر دهید که پس از موفق شدن کاربر دوم، تعداد حدس‌های زده‌شده را با پیام مناسب چاپ کند.
پاسخ:

```

int c = 0;
Console.Write("your month? ");
string month = Console.ReadLine();
Console.Clear();
Console.Write("your guess? ");
string guess = Console.ReadLine();
for (guess = ""; guess != month; guess =
Console.ReadLine())
{
    Console.WriteLine("type your guess");
    c++;
}
Console.WriteLine("you win , month is " +
month);
Console.WriteLine(c + " suspect after
you win");

```

جدول را به کمک هم‌گروهی خود کامل کنید.

فعالیت کارگاهی
ص ۷۰

خروجی برنامه پس از اجرا	قطعه کد
10 12 14	<pre>for (int k = 10; k <= 15; k++) { Console.WriteLine(k); k++;}</pre>
ابتدا متغیر k با مقدار ۱۰ پر می‌شود و چون کمتر از ۱۵ است نمایش داده می‌شود. سپس یک واحد اضافه‌شده (۱۱) و در برگشت حلقه به سمت بالا یک واحد دیگر نیز به آن اضافه می‌شود (۱۲) بنابراین در هر بار گردش ۲ واحد به این متغیر اضافه‌شده است.	

<p>بی‌نهایت Iran چاپ می‌شود این حلقه به خاطر نداشتن شمارنده در واقع به صورت یک حلقه متناوب و بی‌نهایت فعالیت می‌کند. از این حلقه در مسائلی که احتیاج به تکرار بدون پایان داریم استفاده می‌کنیم.</p>	<pre>for (; ;) { Console.ForegroundColor = ConsoleColor.Cyan; Console.WriteLine("Iran");}</pre>
<p>۵ بار Iran زیر هم به رنگ فیروزه‌ای چاپ می‌شود ابتدا متغیر i با مقدار ۱ پر می‌شود و چون کمتر از ۱۰ است یک بار کلمه Iran نمایش داده می‌شود. سپس یک واحد اضافه‌شده (۲) و در برگشت حلقه به سمت بالا یک واحد دیگر نیز به آن اضافه می‌شود (۳) بنابراین در هر بار گردش ۲ واحد به این متغیر اضافه‌شده در واقع متغیر i مقادیر ۱ و ۳ و ۵ و ۷ و ۹ را در خود جای می‌دهد که به تعداد آن‌ها (۵ بار) کلمه Iran نمایش داده می‌شود.</p>	<pre>int i = 1; for (; i<=10; i++) { Console.ForegroundColor = ConsoleColor.Cyan; Console.WriteLine("Iran"); i++;}</pre>

فعالیت منزل
ص ۷۱

برای قطعه کد زیر پس از اجرا و بررسی نتیجه، یک مسئله کاربردی بنویسید. پاسخ: هنرجو می‌تواند یک یا چند مسئله برای این برنامه مثال بزند مانند:

- نمایش مضارب عدد سه کوچک‌تر از ۱۰۰
- بازی هپ، برای مضارب ۳ طراحی شده است.

فعالیت منزل
ص ۷۲

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

پاسخ:

```
string name;
byte count = 0;
byte n;
byte i;
Console.WriteLine("number of students");
n = byte.Parse(Console.ReadLine());
```

```

        Console.Clear();
        Console.WriteLine("name of student");
        for (i = 1; i <= n; ++i)
        {
            name = Console.ReadLine();
            if (name == "masoomeh")
                ++count;
        }

```

```

        Console.WriteLine("tedad ==> {0}", count);

```

- برنامه‌ای بنویسید که چراغ راهنمایی و رانندگی را شبیه‌سازی کند. چراغ راهنمایی و رانندگی ۵ بار از ابتدا تا انتها تکرار شود.

پاسخ:

برای حل این برنامه از ویژگی `foregroundColor` و متدهای `setcursorposition` و `beep` استفاده شده است. در ادامه شرح مختصر این دستورات آمده است.

الف (`foregroundColor`)

از این ویژگی برای رنگ‌آمیزی پس‌زمینه استفاده می‌شود. شکل کلی آن به صورت زیر است.

نام رنگ مورد نظر `ConsoleColor` = `Console.ForegroundColor` با داده شمارشی `ConsoleColor` می‌توانید رنگ مورد نظر خود را انتخاب کنید.

ب (`SetCursorPosition`)

از این متد برای تعیین موقعیت مکان‌نما در صفحه‌نمایش استفاده می‌شود. این متد دارای دو پارامتر است که مقادیر `X` و `Y` نقطه موردنظر خواهند بود. در واقع با استفاده از این پارامترها مختصات نقطه موردنظر را مشخص می‌کنیم. شکل کلی این متد به صورت زیر است.

```

Console.SetCursorPosition(x, y);

```

ج (`beep`)

از این متد برای تولید صدا توسط اسپیکر داخلی رایانه استفاده می‌شود. این متد دارای دو پارامتر به ترتیب فرکانس صدا و مدت‌زمان تولید صدا است. شکل کلی آن به صورت زیر است.

```

Console.Beep(F, T);

```

`F` مقدار عددی فرکانس بر اساس هرتز
`T` مقدار عددی زمان بر اساس میلی‌ثانیه
 کد این فعالیت:

```

for (int i = 0; i < 5; i++)
{
    Console.Clear();
    Console.ForegroundColor =

```

```
ConsoleColor.Gray;
Console.SetCursorPosition(4, 1);
Console.WriteLine("█");
Console.SetCursorPosition(4, 2);
Console.WriteLine("█");
Console.SetCursorPosition(4, 3);
Console.WriteLine("█");
Console.SetCursorPosition(4, 4);
Console.WriteLine("█");
Console.SetCursorPosition(4, 5);
Console.WriteLine("█");
Console.SetCursorPosition(4, 6);
Console.WriteLine("█");
Console.SetCursorPosition(4, 7);
Console.WriteLine("█");
Console.SetCursorPosition(4, 8);
Console.WriteLine("█");
Console.SetCursorPosition(4, 9);
Console.WriteLine("█");
Console.SetCursorPosition(4, 10);
Console.WriteLine("█");
Console.SetCursorPosition(4, 11);
Console.WriteLine("█");
Console.SetCursorPosition(4, 12);
Console.WriteLine("█");
Console.SetCursorPosition(4, 13);
Console.WriteLine("█");
Console.SetCursorPosition(12, 1);
Console.WriteLine("█");
Console.SetCursorPosition(12, 2);
Console.WriteLine("█");
Console.SetCursorPosition(12, 3);
Console.WriteLine("█");
Console.SetCursorPosition(12, 4);
Console.WriteLine("█");
Console.SetCursorPosition(12, 5);
Console.WriteLine("█");
Console.SetCursorPosition(12, 6);
Console.WriteLine("█");
Console.SetCursorPosition(12, 7);
Console.WriteLine("█");
Console.SetCursorPosition(12, 8);
Console.WriteLine("█");
Console.SetCursorPosition(12, 9);
```



```

Console.WriteLine("█");
Console.SetCursorPosition(12, 10);
Console.WriteLine("█");
Console.SetCursorPosition(12, 11);
Console.WriteLine("█");
Console.SetCursorPosition(12, 12);
Console.WriteLine("█");
Console.SetCursorPosition(12, 13);
Console.WriteLine("█");
Console.SetCursorPosition(8, 14);
Console.WriteLine("█");
Console.SetCursorPosition(8, 15);
Console.WriteLine("█");
Console.SetCursorPosition(8, 16);
Console.WriteLine("█");
Console.SetCursorPosition(8, 17);
Console.WriteLine("█");
Console.SetCursorPosition(8, 18);
Console.WriteLine("█");
Console.SetCursorPosition(8, 19);
Console.WriteLine("█");
Console.SetCursorPosition(8, 20);
Console.WriteLine("█");
Console.SetCursorPosition(8, 21);
Console.WriteLine("█");
Console.SetCursorPosition(8, 22);
Console.WriteLine("█");
Console.SetCursorPosition(4, 1);
Console.WriteLine("██████████");
Console.SetCursorPosition(4, 13);
Console.WriteLine("██████████");
Console.SetCursorPosition(4, 5);
Console.WriteLine("██████████");
Console.SetCursorPosition(4, 9);
Console.WriteLine("██████████");
Console.Beep(37, 100);
Console.SetCursorPosition(5, 2);
Console.ForegroundColor =
ConsoleColor.Red;
Console.WriteLine("██████████");
Console.SetCursorPosition(5, 3);
Console.WriteLine("██████████");
Console.SetCursorPosition(5, 4);
Console.WriteLine("██████████");

```

```

        Console.WriteLine();
        Console.Beep(37, 3000);
        Console.SetCursorPosition(5, 2);
        Console.ForegroundColor =
ConsoleColor.Black;
        Console.WriteLine("██████████");
        Console.SetCursorPosition(5, 3);
        Console.WriteLine("██████████");
        Console.SetCursorPosition(5, 4);
        Console.WriteLine("██████████");
        Console.WriteLine();
        Console.ForegroundColor =
ConsoleColor.Yellow;
        Console.SetCursorPosition(5, 6);
        Console.WriteLine("██████████");
        Console.SetCursorPosition(5, 7);
        Console.WriteLine("██████████");
        Console.SetCursorPosition(5, 8);
        Console.WriteLine("██████████");
        Console.WriteLine();
        Console.Beep(37, 2000);
        Console.ForegroundColor =
ConsoleColor.Black;
        Console.SetCursorPosition(5, 6);
        Console.WriteLine("██████████");
        Console.SetCursorPosition(5, 7);
        Console.WriteLine("██████████");
        Console.SetCursorPosition(5, 8);
        Console.WriteLine("██████████");
        Console.WriteLine();
        Console.ForegroundColor =
ConsoleColor.Green;
        Console.SetCursorPosition(5, 10);
        Console.WriteLine("██████████");
        Console.SetCursorPosition(5, 11);
        Console.WriteLine("██████████");
        Console.SetCursorPosition(5, 12);
        Console.WriteLine("██████████");
        Console.Beep(37, 7000);
    }

```

فعالیت
تکمیلی

آیا شما می‌توانید تعداد خطوط برنامه را کاهش دهید؟ از چه روشی استفاده می‌کنید؟
برنامه را مجدداً با روش جدید بنویسید.

فعالیت گروهی
ص ۷۳

با کمک هم گروهی خود دو مثال دیگر برای حلقه نامعین بنویسید.
پاسخ:

- ۱- برنامه اعلام شماره تا زمان اعلام شماره ۱۰۰۰.
- ۲- برنامه مطالعه صفحات کتاب درسی تا هنگام خسته شدن.

کارگاه ۶: به کارگیری حلقه while

پاسخ به فعالیت‌ها

تکمیل کارگاه
ص ۷۴

- آکولادهای حلقه را حذف کرده، برنامه را اجرا کنید.
پاسخ:

با اجرای برنامه عدد ۱ به تعداد بی‌نهایت نمایش داده می‌شود. زیرا فقط یک خط در حلقه اجرا می‌شود. مقدار اولیه حلقه یک بوده و دستور افزایش شمارنده حلقه اجرا نمی‌شود.

- دستورات داخل حلقه را به یک دستور تبدیل کنید.
پاسخ:

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

- پس از شرط دستور while سعی کالن قرار دهید. چه تغییری در نتیجه رخ می‌دهد؟
پاسخ:

برنامه هیچ خروجی ندارد چون شرط دستور while همواره برقرار است یک حلقه بی‌نهایت اجرا خواهد شد. برای خروج کلیدهای Ctrl+Breach را فشار دهید. پس از انجام این مرحله این تغییر حذف شود که برنامه دچار حلقه بی‌نهایت نشود.

- دستور مربوط به افزایش شمارنده حلقه (i++) را در قطعه کد مرحله ۳ حذف کرده برنامه را اجرا کنید.
پاسخ:

در صورتی که دستور افزایش شمارنده حلقه حذف شود مقدار اولیه حلقه هیچ‌گاه افزایش پیدا نکرده و عدد ۱ بی‌نهایت بار نمایش داده می‌شود.

فعالیت کارگاهی
ص ۷۴

با کمک هم گروهی خود معادل هر یک از قطعه کدها را با دستور خواسته شده بنویسید و آن‌ها را اجرا کنید.
پاسخ:

نتیجه	قطعه برنامه با حلقه while	قطعه برنامه حلقه for
چاپ اعداد ۱۰۰ تا ۱۰۰۰ به ترتیب	<pre>int i = 100; while (i <= 1000) { Console.WriteLine(i); i++; }</pre>	<pre>for (int i = 100; i <= 1000;i++) Console.WriteLine(i);</pre>
چاپ اعداد زوج ۲۰ تا ۰ به صورت نزولی	<pre>int i = 20; while (i >=0) { Console.WriteLine(i); i=i-2; }</pre>	<pre>for (int i = 20; i >= 0;i-=2) Console.WriteLine(i);</pre>

فعالیت منزل
ص ۷۵

عملکرد قطعه کد زیر چیست ؟ این برنامه را با دستور while بنویسید.

پاسخ:
ابتدا تعریف متغیرها :

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

سپس قطعه برنامه با دستور while

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

- اولین و آخرین مقدار چاپ شده دستورات زیر را به دست آورید :

آخرین مقدار در خروجی	اولین مقدار در خروجی	دستور
۰	۹۸	<pre>int i = 99; while (i>=1) Console.WriteLine(--i);</pre>
<p>ابتدا متغیر i با ۹۹ مقداردهی شده است، شرط برقرار است اما هنگام نمایش مقدار i یکی از آن کم می‌شود و عدد ۹۸ نمایش داده می‌شود. آخر حلقه نیز مقدار i یک می‌شود و به همان علت عدد 0 نمایش داده می‌شود.</p>		
۱۰۰۱	۱۰۱	<pre>int i = 100; while (i <= 1000) Console.WriteLine(++i);</pre>
<p>ابتدا مقدار متغیر i، ۱۰۰ می‌شود و چون کمتر از ۱۰۰۰ است حلقه اجرا شده و ابتدا مقدار ۱۰۱ نمایش داده می‌شود و در آخر حلقه که مقدار ۱۰۰۰ می‌آید یکی اضافه‌شده و ۱۰۰۱ به نمایش در خواهد آمد.</p>		

فعالیت
تکمیلی

جدول زیر را تکمیل کنید.

دستور	اولین مقدار در خروجی	آخرین مقدار در خروجی
<pre>byte i=0 ; byte n=10 ; while (true) { Console.WriteLine(i); ++i; if(i>n) break ; } </pre>		
<pre>int n = 356; while (n != 0) { Console.WriteLine(n % 10); n = n / 10; } </pre>		

کارگاه ۷ : یافتن بزرگ‌ترین مقدار

پاسخ به فعالیت‌ها

- متغیرهای استفاده‌شده در قطعه کد زیر را اعلان کرده، قطعه کد زیر را تکمیل نموده و برنامه را اجرا کنید.

تکمیل کارگاه
ص ۷۵

پاسخ:

```
int hour, max;
Console.ForegroundColor = ConsoleColor.Blue;
Console.WriteLine("\nEnter 0 for Exit");
Console.Write("enter hour: ");
hour = int.Parse(Console.ReadLine());
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 داخل حلقه while بزرگ‌تر از صفر است و با وارد کردن یکی از اعداد منفی یا صفر شرط خروج از حلقه برقرار و از برنامه خارج می‌شود.
- برنامه را طوری تغییر دهید که کمترین و بیش‌ترین ساعت کاری را نمایش بدهد.

پاسخ:

برای به دست آوردن مقدار بزرگ‌تر و کوچک‌تر از میان تعدادی عدد راه‌حل ارائه‌شده این است که مقدار اول را هم بزرگ‌تر و هم کوچک‌تر در نظر بگیریم و باقیمانده مقادیر را با آن‌ها مقایسه کنیم. بنابراین در کد نوشته‌شده مقدار اولین ساعت خوانده‌شده و به عنوان بزرگ‌ترین و کوچک‌ترین مقدار در متغیرهای min و max ذخیره‌شده است و پس‌از آن تا زمانی که مقدار واردشده بزرگ‌تر از صفر است، مقادیر خوانده‌شده با همین دو متغیر مقایسه شده و در صورت برقراری شروط نوشته‌شده مقادیر این متغیرها تغییر می‌کند.

```
int hour, min, max;
Console.ForegroundColor = ConsoleColor.Blue;
Console.WriteLine("\nEnter 0 for Exit");
Console.Write("enter hour: ");
hour = int.Parse(Console.ReadLine());
max = hour;
min = hour;
while (hour > 0)
{
    if (hour > max)
        max = hour;
    else if (hour < min)
```

```

min = hour;
Console.WriteLine("\nEnter 0 for Exit");
Console.Write("enter hour: ");
hour = int.Parse(Console.ReadLine());
}
Console.WriteLine("max= {0}", max);
Console.WriteLine("min= {0}", min);

```

برنامه نوشته شده در کارگاه ۳ را با استفاده از حلقه while نوشته، اجرا کنید.

```

int score = 5;
byte guess, month;
Console.Write("Enter your month: ");
month = byte.Parse(Console.ReadLine());
Console.Clear();
byte i = 1;
while (i <= 5)
{
    Console.Write("your guess? ");
    guess = byte.Parse(Console.ReadLine());
    i++;
    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 ! ");
    }
}
}

```

فعالیت کارگاهی

ص ۷۶

کارگاه ۸ : کاربرد حلقه های متداخل

در صورتی که دو یا چند حلقه را پشت سر هم بنویسیم حلقه تودرتو ایجاد کرده ایم. به تکه برنامه زیر نگاه کنید .

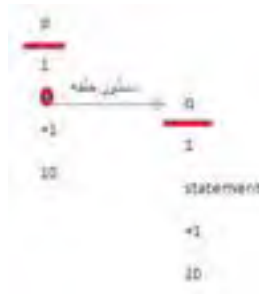
```
for (byte i = 0; i <= 10; ++i)
```

```
for (byte j = 0; j <= 10; ++j)
    Console.WriteLine("*");
```

درواقع ما در یک حلقه for می‌توانیم در بدنه آن یک حلقه دیگر را تعریف کنیم. البته چنین تعریفی نیز در خصوص حلقه نامعین while صادق است.

شکل زیر مفهوم حلقه تودرتو را برای حلقه معین for نشان می‌دهد. در این شکل می‌بینید که بجای دستور حلقه p یک حلقه دیگر یعنی حلقه q اجرا می‌شود.

به‌طور کلی حلقه‌های تودرتو (بدون توجه به معین بودن یا نامعین بودن) را می‌توان به دو دسته کلی تقسیم کرد.



۱- حلقه‌های تودرتوی مستقل

تمام قسمت‌های حلقه (مقدار اولیه - مقدار نهایی - گام حلقه - دستور حلقه) مستقل، مشخص و معلوم هستند.

```
for(byte i=0;i<=10;++i)
for(byte j=0;j<=10;++j)
Console.WriteLine("*");
```

در این نوع حلقه‌ها به راحتی می‌توان تعداد تکرار دستورات را محاسبه کرد.

تعداد تکرار حلقه = ۱ + مقدار اولیه - مقدار نهایی

کافی است از روش محاسبه تعداد تکرار حلقه‌های مستقل تکی تعداد تکرار حلقه‌ها را به صورت مجزا حساب کرده سپس عددهای به دست آمده را در یکدیگر ضرب کنیم.

تعداد کل تکرارها = تعداد تکرار حلقه اول * تعداد تکرار حلقه دوم ... * * *

۲- حلقه‌های تودرتوی غیرمستقل (وابسته)

در این نوع حلقه‌ها، حلقه‌های داخلی به حلقه‌های بیرونی وابسته هستند درواقع یکی از قسمت‌های حلقه معلوم و مشخص نیست.


```
for (byte p = 0; p <= 10; ++p)
for (byte q = 0; q <= p; ++q)
Console.WriteLine("*");
```

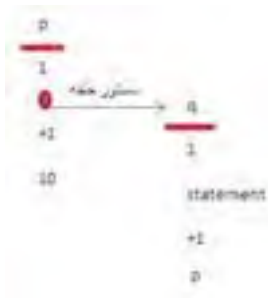
مشاهده می‌کنیم که حلقه دوم یعنی حلقه q در بخش نهایی به حلقه p وابسته است. تأثیری که این وابستگی دارد این است که دیگر نمی‌توان مقدار گردش حلقه دوم را به صورت صحیح و با استفاده از فرمول محاسبه کرد.

```
for (byte p=1 ; p<=5 ; ++p)
for(byte q=1 ; q<=p;++q)
    Console.WriteLine("{0}",q);
```

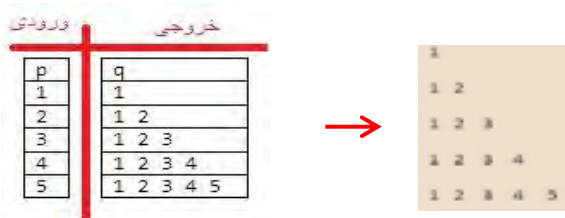
بنابراین برای محاسبه تعداد گردش یا به دست آوردن خروجی این‌گونه حلقه‌ها باید از جدول درستی که قبلاً توضیح دادیم بهره ببریم.

```
for (byte p=1 ; p<=5 ; ++p)
for(byte q=1 ; q<=p;++q)
    Console.WriteLine("{0}",q);
```

مثال: خروجی تکه برنامه زیر چیست ؟



برای به دست آوردن خروجی این برنامه یک جدول درستی با همان توضیحات قبلی ترسیم می‌کنیم. می‌بینید که تعداد گردش حلقه دوم یعنی حلقه q به حلقه p وابسته شده است. برای مثال هنگامی که p مقدار ۳ را دریافت کرده حلقه q نیز تا شماره ۳ حرکت کرده است. در شکل خروجی نهایی این تکه برنامه را مشاهده می‌کنیم.



پاسخ به فعالیت‌ها

سوال
ص ۷۸

برای تنظیم نمایش جدول ضرب به‌جای جانگهدار از متد `SetCursorPosition` استفاده کنید.

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

فعالیت گروهی
ص ۷۸

با کمک هم‌گروهی خود بررسی کنید دستور `Console.WriteLine` در قطعه کد زیر چند بار اجرا می‌شود؟

```
for (int j = 1; j <= 4; j++)
    for (int k = 1; k <= j; k++)
        Console.WriteLine("Iran");
```

در این قطعه کد مقدار متغیر k تا زمانی اضافه می‌شود که بیشتر از مقدار j نباشد، در اولین مرحله که مقدار k برابر ۱ است فقط یک‌بار حلقه داخلی اجرا می‌شود، وقتی حلقه بیرونی برای دومین بار اجرا شود، مقدار حلقه بیرونی و متغیر k برابر ۲ شده و حلقه داخلی هم دو بار اجرا می‌شود. همین‌طور ادامه پیدا کرده تا حلقه بیرونی ۴ شده و حلقه داخلی هم چهار بار اجرا شود. در این کنجکاوی هنرآموز از هنرجو جدول `Trace` بخواند تا هنرجو بتواند وابستگی حلقه‌ها را بهتر درک کند.

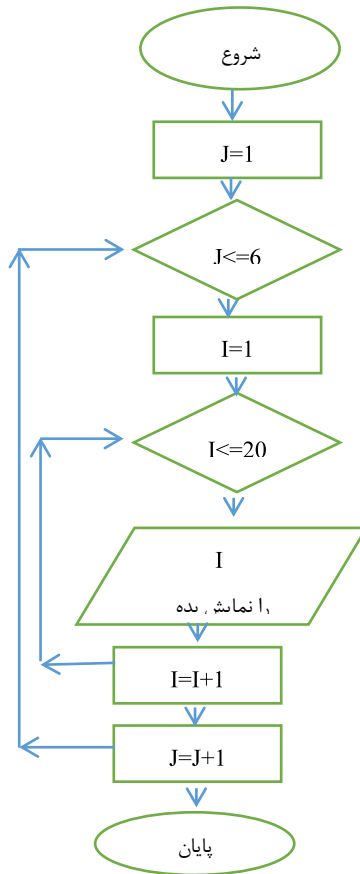
متغیر j	متغیر k	خروجی
1	1	Iran
2	1	Iran
	2	Iran
3	1	Iran
	2	Iran
	3	Iran
4	1	Iran
	2	Iran

		3	Iran	
		4	Iran	

فعالیت منزل
ص ۷۸

سیستم نوبت دهی مطب دندان پزشکی را طوری تغییر دهید که برای یک هفته که شامل شش روز کاری است نوبت دهی انجام دهد.

```
for (byte j = 1; j <= 6; j++)
    for (byte i = 1; i <= 20; i++)
        Console.WriteLine(i);
```



کارگاه ۹: توسعه حلقه متداخل

پاسخ به فعالیت‌ها

تکمیل کارگاه
ص ۷۸

- جدول روبه‌رو را تکمیل کنید. برنامه را اجرا کنید آیا با خروجی‌های جدول شما مطابقت دارد؟

پاسخ:

تعداد اسکناس‌های ۲ هزار تومانی	تعداد اسکناس‌های ۵ هزار تومانی
۵	۸
۱۰	۶
۱۵	۴
۲۰	۲

بله- دقیقاً همان خروجی‌هایی را تولید خواهد کرد که در جدول به دست آورده بودیم. البته خودپرداز بر اساس موجودی اسکناس‌ها پرداخت را انجام خواهد داد. اما در اینجا اولین مقادیر X و Y به دست آمده را استخراج خواهد کرد یعنی همان ردیف اول جدولی که شما پر کرده‌اید.

مقدار نهایی X مقدار ۲۵ در نظر گرفته شده است، زیرا بر اساس رابطه $25 * 2000 = 50000$ خواهد شد و مقدار نهایی Y مقدار ۱۰ در نظر گرفته شده است، زیرا بر اساس رابطه $50000 = 10 * 5000$ خواهد شد.

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

پاسخ:

برای به دست آوردن جواب این سؤال ابتدا باید تعداد تکرار حلقه X و سپس تعداد تکرار حلقه Y را به دست آورده و سپس این مقادیر را در یکدیگر ضرب کنیم.

$$X \text{ تکرار حلقه} = 25 - 1 + 1 = 25$$

$$Y \text{ تکرار حلقه} = 10 - 1 + 1 = 10$$

و در نهایت تعداد کل تکرارها برابر است با $25 * 10 = 250$

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

```
for( int x=1 ; x <= 25 ; x++ )
for( int y=1 ; y <= 10 ; y++ )
for ( int z=1 ; z <= 5 ; z++ )
if ( 2000*x + 5000*y + 10000*z == 50000 )
Console.WriteLine("X={0},Y={1},Z={2}",x,y,z) ;
```

- برنامه را طوری تغییر دهید که وجه نقد درخواستی کاربر را از ورودی دریافت کند.

```
int p ;
p= int.Parse(Console.ReadLine());
for( int x=1 ; x <= p/2 ; x++ )
for( int y=1 ; y <= p/5 ; y++ )
if ( 2000*x + 5000*y == p)
Console.WriteLine ("X={0},Y={1}",x,y) ;
```

- خودپرداز پرداخت اسکناس را بر اساس کمترین تعداد اسکناس انجام می‌دهد. برنامه‌ای بنویسید که وجه درخواستی کاربر را بگیرد و پرداخت خودپرداز را تعیین کند.

پاسخ:

```
byte s1 = 0;
byte s2 = 0;
byte s3 = 0;
int p;
p = int.Parse(Console.ReadLine());
if (p >= 50000)
{
    s1 = (byte)(p / 50000);
    p = p - (s1 * 50000);
    if (p >= 10000)
    {
        s2 = (byte)(p / 10000);
        p = p - (s2 * 10000);
        s3 = (byte)(p/5000);
    }
    Console.WriteLine(" {0} --> 50000 \n {1} --> 10000 \n {2} -->
5000 ", s1, s2, s3);
}
else
{
    if (p < 50000)
    {
        s1 = (byte)(p / 10000);
        p = p - (s1 * 10000);
        if (p >=5000)
        {
            s2 = (byte)(p / 5000);
            p = p - (s2 * 5000);
        }
        Console.WriteLine(" {0} --> 10000 \n {1} --> 5000 ", s1, s2);
    }
}
```

- می‌خواهیم برنامه‌ای بنویسیم که تعداد گروه‌های حاضر در بازی دوره‌ای را بگیرد و جدول بازی‌های این گروه را در خروجی نمایش بدهد. (نام گروه‌ها را با حروف **A** و **B** و **C** و نام‌گذاری کنید.)

پاسخ:

```
Char p ;
Char q ;
int ch;
for (p = 'A'; p <= 'E'; p++)
{
    ch=p+1;
    for (q = (char)ch; q <= 'E'; q++)
        Console.WriteLine("{0} --> {1}",p,q);
}
```

واحد یادگیری ۴

شایستگی کار با آرایه و داده‌های شمارشی

مقدمات تدریسی

الف) مفاهیم کلیدی

مفاهیم کلیدی		
آرایه	عنصر	داده شمارشی
پیمایش آرایه	جستجوی خطی	رشته
جستجوی دودویی	کاراکتر	کلاس آرایه

ب) مراحل انجام کار واحد یادگیری ۴

مراحل انجام کار	
۱	اعلان آرایه
۲	دسترسی به عناصر آرایه
۳	به‌کارگیری متدهای آماده روی آرایه
۴	به‌کارگیری داده‌های شمارشی

ج) بودجه‌بندی

زمان‌بندی واحدهای یادگیری:

ردیف	اهداف توانمندسازی	سهم زمان
۱	درک مفهوم آرایه تعریف یک آرایه - پر کردن خانه‌های آرایه - دسترسی به عنصر خاصی در آرایه	۷ ساعت
۲	درک و توانایی دسترسی به خانه‌های آرایه دسترسی به همه عناصر یا یک عنصر خاص از آرایه - نمایش مقدار یا مقادیر عناصر آرایه	۷ ساعت
۳	درک مفهوم جستجو و ساخت و پیاده‌سازی روش‌های جستجو شناخت روش‌های جستجوی در آرایه - ایجاد یک روش جستجو - به‌کارگیری یکی از روش‌های جستجو در حل مسئله	۱۱ ساعت
۴	به‌کارگیری داده‌های شمارشی تعریف و به‌کارگیری داده‌های شمارشی و حل مسئله با استفاده از آن	۵ ساعت
مجموع زمان		۳۰ ساعت

طرح درس روزانه:

طرح درس روزانه (هفتگی) پیشنهادی				
درس: کار با آرایه و داده‌های شمارشی			کلاس: یازدهم	
پیام جلسه (هدف کلی): هنرجو بتواند مسئله آرایه‌ای را تشخیص داده و آن را با آرایه انجام دهد.				
زمان فیزیکی	فعالیت‌ها		اهداف یادگیری	
مدت (دقیقه)	کار هنرجو	کار هنرآموز	طبقه هدف: حیطه عاطفی / شناختی / روانی - حرکتی	فعالیت
۳۰	مشارکت در پاسخگویی و تعامل	تعریف آرایه - اعلان آرایه - معرفی شماره عناصر آرایه	سنجش میزان آگاهی هنرجویان از کد نویسی و آشنایی با مفهوم آرایه	ارزشیابی رفتار ورودی
۳۰	یک روند نما برای آرایه ترسیم کند.	با استفاده از آرایه یک مسئله را حل کند.	نمایش فیلم	ایجاد انگیزه
۸۰	هنرجویان مطالب پایه را می‌آموزند.	مسئله‌ای را با استفاده از آرایه حل کند. روش‌های جستجو را بیان کرده و کد نویسی لازم را انجام دهد.	توضیح کامل مفاهیم کلیدی (دانشی) و ایجاد علاقه و انگیزه در هنرجویان (بینشی)	ارائه مفاهیم کلیدی (توضیح هنرآموز)
۸۰	با توجه به فعالیت‌های کارگاهی کتاب، هنرجویان در گروه‌های خود، گزینه‌های دیگر را با آزمون و خطا می‌آموزند.	هنرآموز توضیح کلی در مورد فعالیت کارگاهی به هنرجو داده و از آن‌ها می‌خواهد به صورت گروهی به انجام آن بپردازند.	هنرجو باید توانایی کار با آرایه را داشته باشد و کارگاه عملی و فعالیت کارگاهی را انجام دهد.	فعالیت کارگاهی (تمرین هنرجویان)
۵۰	هنرجویان در گروه‌های خود سعی می‌کنند با نهایت دقت و سرعت با ایجاد خلاقیت به انجام فعالیت کارگاهی بپردازند.	از هنرجویان خواسته می‌شود در گروه‌های تعیین شده به انجام فعالیت‌های کارگاهی بپردازند. قبل از شروع به کار هنرجویان، ملاک‌های ارزیابی تمرین‌ها به آن‌ها داده می‌شود.	طرح یک مسئله که با استفاده از حلقه حل می‌شود.	ارزیابی فعالیت‌ها ارائه تمرین
۷۰	توجه و دقت و پرسش و پاسخ، انجام فعالیت‌های گروهی و ارائه راه‌حل‌های جدید برای مسائل.	با توجه به مفاهیم بیان شده، بخشی از فعالیت‌های کارگاهی را برای هنرجویان شرح دهید.	هنرجو باید مسئله را شناخته و آرایه مناسب را انتخاب کرده و روند نمای مربوط به آن را ترسیم کرده و به کد نویسی بپردازد.	ارائه نکات تکمیلی (جمع‌بندی)

طرح درس روزانه (هفتگی) پیشنهادی			
کلاس: یازدهم		درس: کار با آرایه و داده‌های شمارشی	
پیام جلسه (هدف کلی): هنرجو بتواند مسئله آرایه‌ای را تشخیص داده و آن را با آرایه انجام دهد.			
اهداف یادگیری	فعالیت‌ها	زمان فیزیکی	
ارزشیابی شایستگی (ارزشیابی پایانی)	این ارزشیابی در دوشاخه انفرادی و گروهی انجام می‌شود: از هنرجویان خواسته می‌شود در گروه‌های تعیین شده به انجام پروژه این بخش از واحد کار بپردازند. نمره‌ای به کار گروهی هنرجویان داده می‌شود. آزمون برای هنرجویان به صورت انفرادی برگزار می‌شود. نمره گروهی در نمره انفرادی افراد تأثیرگذار است. نکته: قبل از شروع به کار، ملاک‌های ارزیابی پروژه به آن‌ها داده می‌شود.	۸۰	هنرجویان در گروه‌های خود سعی می‌کنند در مدت‌زمان تعیین شده با نهایت دقت و سرعت با ایجاد خلاقیت به انجام پروژه بپردازند.
ابزارهای مورد نیاز	ویدئو پروژکتور، رایانه، تخته آموزشی، برنامه VS		

د) فیلم‌ها و پرونده‌های پیوست

شماره فیلم	هدف
111h12	تعریف آرایه
111h13	استفاده از آرایه در حل مسائل
111h14	ویژگی Length
111h15	مرتب‌سازی آرایه
111h16	جستجوی خطی
111h17	جستجوی دودویی
111h18	کلاس Array

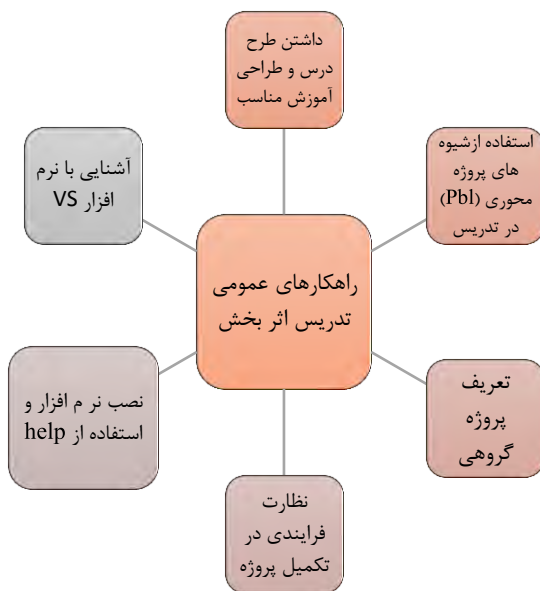
نکته: برای انجام فعالیت‌های تکمیلی می‌توانید از کدهای نوشته‌شده آماده و موجود در «پرونده پیوست فعالیت‌های تکمیلی» استفاده کنید.

ه) ورود به بحث

بارها اتفاق افتاده است که شما احتیاج به ذخیره داده‌هایی برای استفاده در حال و آینده داشته‌اید. از هنرجوی خود بپرسید که آیا تا به حال به فهرستی از داده برخورد کرده است؟ آیا در اطراف خود داده‌های هم نوعی را دیده است؟ شاید بزرگ‌ترین چالش این فصل درک مفهوم پلاک و اندیس خانه‌های آرایه است. بهتر است خیابان یا کوچه‌ای که هنرجوی شما در آن زندگی می‌کند را مثال بزنید. مثال‌های متنوعی را برای هنرجویان خود مطرح کنید تا در حین تدریس این مشکل را به حداقل برسانید. مثلاً به آن‌ها بگویید فهرستی از اسامی دوستان هم‌محل‌های خود تهیه کنند.

به‌منظور تعیین سطح دانش‌آموزان می‌توانید سؤالاتی از این دست طرح کنید:

- سؤال ۱: به نظر شما چگونه می‌توانیم قد دانش‌آموزان یک کلاس را ثبت کنیم؟
- سؤال ۲: کدام یک از شما قبلاً تعدادی اسم را در کنار هم نوشته‌اید؟ مثال بزنید:
- سؤال ۳: آیا تا به حال در یک کوچه به دنبال یک شخص که فقط نام خانوادگی او را می‌دانید گشته‌اید؟
- سؤال ۴: آیا تا به حال فهرستی از اسامی و داده‌ها را دیده‌اید؟ مثال بزنید:
- سؤال ۵: آیا تا به حال در بین تعدادی از اسامی به دنبال اسم خاصی گشته‌اید؟
- سؤال ۶: چه روشی برای به خاطر سپردن اسامی ماه‌های سال پیشنهاد می‌کنید؟
- سؤال ۷: چند نمونه از کاربردهای فهرست را نام ببرید؟



آرایه

در برنامه‌نویسی یاد گرفتیم که یک متغیر را تعریف کرده و سپس آن را مقداردهی کنیم. اما مشکلی که پیش می‌آید این است که مقدار دوم باعث از بین رفتن مقدار اول و به همین ترتیب مقدار سوم باعث از بین رفتن مقدار دوم می‌شود و این روال همچنان ادامه دارد. (شکل زیر را ببینید)



در این شکل مقدار نهایی متغیر A عدد ۱۷ خواهد بود و مقادیر قبل از ۱۷ همگی از بین خواهند رفت. حال این سؤال پیش می‌آید که اگر بخواهیم داده‌های قبلی این متغیر را نگهداری کنیم چه کار باید انجام دهیم؟

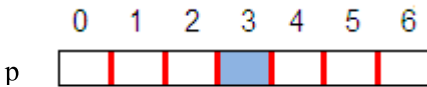
مثال: برنامه‌ای بنویسید که ۱۰۰ عدد را به عنوان دمای شهر بخواند، ابتدا اعداد مثبت (دمای بالای صفر) و سپس اعداد منفی (دمای زیر صفر) را نمایش دهد.

در صورتی که از روش‌های قبلی استفاده کنید برای ذخیره این اعداد به صورت جداگانه به ۱۰۰ متغیر احتیاج خواهید داشت. شاید بگویید از حلقه‌ها استفاده می‌کنیم. آیا می‌توانید داده‌ها را نگهداری کنید؟ چه مشکلاتی برای حل این‌گونه مسائل خواهید داشت؟ باید متغیری داشته باشیم که ۱۰۰ عدد را در خود نگهداری کند. یعنی متغیری که ۱۰۰ فضای خالی و هم‌نوع و هم‌اندازه در اختیار شما بگذارد. به این‌گونه متغیرها آرایه می‌گویند.

فرض کنید یک‌تکه مستطیل به نام P دارید و آن را به قسمت‌های مساوی تقسیم می‌کنیم. (شکل زیر)

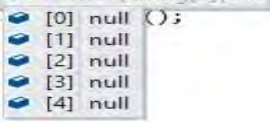


مستطیل مشکی‌رنگ را با خطوط قرمز به قسمت‌های مساوی تقسیم کرده‌ایم. راه‌حل دسترسی به خانه رنگی شده شماره‌گذاری تمام خانه‌ها است.



حال به شکل جدید نگاه کنید. در تمام زبان‌های برنامه‌نویسی رایج شماره خانه‌های آرایه که به آن‌ها عناصر آرایه می‌گویند از صفر شروع می‌شود. پس برای دسترسی به خانه شماره ۴ از آرایه p باید اندیس ۳ را در نظر بگیریم (p[3]). می‌توانیم برای انجام عملیات بر روی آرایه از حلقه‌های معین و نامعین (که عملیات شمارش را انجام می‌دهند)

```
static void Main(string[] args)
{
    string[] name;
    name = new string[5];
    name[0] = "ali";
    Console.WriteLine(name);
}
```



استفاده کنیم. به تکه برنامه زیر دقت کنید. آرایه computer دارای ۵ خانه هست که هر کدام از این خانه‌ها می‌تواند یک‌رشته را در خود جای دهد، پس نوع آرایه رشته‌ای است.

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

```
long[] price = new long[5];
```

چون خانه‌های آرایه در کنار هم قرار دارند، بنابراین می‌توان با استفاده از حلقه‌ها بر روی آن‌ها عملیات خواندن، نوشتن یا پردازش را انجام داد.

```
for (int i = 0; i <= 4; i++)
{
    computer[i] = Console.ReadLine();
    price[i] = int.Parse(Console.ReadLine());
}
```

یا

```
for (byte i = 0; i <= 6; ++i)
{
    Console.WriteLine(week[i]);
    ++i;
}
```

یا

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

شیوه و الگوی پیشنهادی

از هنجرویان خود بخواهید که اسامی دو نفر از همکلاسی‌های خود را یادداشت کنند. حالا دوباره بگویید دو نفر دیگر را یادداشت کنند، این کار را برای ۱۰ هنرجو انجام دهید. از آن‌ها بپرسید راهکار جدیدی برای این کار سراغ دارند؟ حتماً

می‌گویند نام ۱۰ نفر را باهم وارد کنیم. حالا از آن‌ها بخواهید که به هر کدام از این اسامی یک شماره اختصاص بدهند. به هنرجویان بگویید نفر شماره ۳ و ۵ را نام ببرند. از هنرجویان بخواهید در بین اسامی نوشته‌شده نام علی را پیدا کنند و شماره خانه‌اش را بنویسند. این بار از آن‌ها بخواهید تا اسامی را مرتب کنند و بار دیگر اسامی را از انتها به ابتدا یادداشت کنند.

مشکلات متداول در فرآیند یاددهی - یادگیری:

شاید بتوان گفت بزرگ‌ترین مشکل درک مفهوم اندیس خانه‌های آرایه است، چراکه اندیس آرایه از شماره صفر شروع می‌شود. درک هنرجویان بر اساس پلاک خانه‌هایی است که در آن‌ها زندگی می‌کنند که از شماره ۱ شروع می‌شوند و این می‌تواند یک چالش برای بیان آرایه باشد.

پاسخ به فعالیت‌ها

روند نمایی رسم کنید که اسامی پنج هنرجو را دریافت کرده سپس اسامی را از آخر به اول نمایش دهد. برنامه این الگوریتم را به زبان سی شارپ بنویسید.
پاسخ:

۰- شروع

۱- خواندن n1,n2,n3,n4,n5

۲- نمایش n5,n4,n3,n2,n1

۳- پایان

در این فعالیت مشاهده می‌کنیم که به تعداد اسامی باید دستورات ورودی و خروجی بکار ببریم. نکته این سؤال در این است که از حلقه for نیز نمی‌توان استفاده کرد. در چنین مسئله‌هایی بهترین روش استفاده از فهرست است.

```
string n1, n2, n3, n4, n5;  
n1 = Console.ReadLine();  
n2 = Console.ReadLine();  
n3 = Console.ReadLine();  
n4 = Console.ReadLine();  
n5 = Console.ReadLine();  
Console.WriteLine("{0}{1}{2}{3}{4}", n5, n4, n3, n2,  
n1);
```

فعالیت کارگاهی

ص ۸۲

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

فعالیت کارگاهی

ص ۸۳

```
static void Main(string[] args)
{
    string[] name;
    name = new string[5];
    name[0] = "ali";
    Console.WriteLine(name);
}

```

فعالیت کارگاهی
ص ۸۴

- آرایه‌ای به نام months برای نگهداری اسامی ماه‌های سال تعریف کنید.
 - آرایه‌ای به نام salary برای نگهداری حقوق ۳۰۰ کارمند تعریف کنید.
 - آرایه‌ای به نام vowels برای نگهداری حروف صدا دار انگلیسی تعریف کنید.
 - آرایه‌ای به نام average برای نگهداری معدل هنرجویان کلاس تعریف کنید.
 - آرایه‌ای به نام lamp برای نگهداری وضعیت روشن و خاموش بودن پنج لامپ تعریف کنید.
- پاسخ:

```
1-string [ ] months = new string[12] ;
2-long [ ] salary = new long [300];
3-char [ ] vowels = new char [5];
4-float [ ] average = new float [30];
5-byte [ ] lamp = new byte [5];

```

کنجکاوی
ص ۸۴

چرا برای مراجعه به خانه سوم آرایه، از اندیس ۲ استفاده شده است؟
پاسخ: چون اندیس‌های خانه‌های آرایه از صفر شروع می‌شوند.

کارگاه ۱ تعریف و مقداردهی آرایه

پاسخ به فعالیت‌ها:

تکمیل کارگاه
ص ۸۵

- برای ذخیره کردن اسامی قطعات دیگر، چه تغییری در کد باید بدهیم؟
پاسخ:
ابتدا باید تعداد خانه‌های آرایه را زیاد کنیم و سپس آن‌ها مقداردهی کنیم.
- برای نگهداری قیمت قطعات رایانه، آرایه price را متناظر با آرایه نام قطعات اعلان کنید.
پاسخ:

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

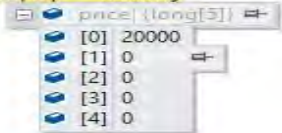
- برنامه را با کلید F10 تا رسیدن به دستور مقاردهی price[2] اجرا کنید و محتوای آرایه price را مشاهده کنید.

پاسخ :

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

computer[0] = "cpu";
computer[1] = "main";
computer[2] = "ram";

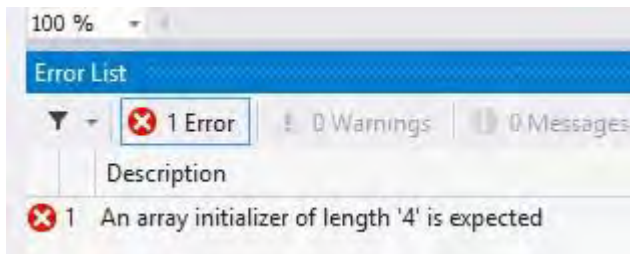
price[0] = 20000;
price[1] = 15000;
```



- دستور زیر را جایگزین کد مرحله ۹ کنید. به جای عدد ۴ عدد ۵ را قرار دهید. خطای رخ داده چیست؟

```
string[] computer= new string[4]{"CPU",
"MainBoard", "RAM", "HDD", "DVD"};
```

پاسخ: در صورتی که تعداد خانه‌های آرایه نسبت به مقادیر وارد شده کمتر باشد برنامه خطا نشان خواهد داد.



- دستوری برای تغییر مقدار عنصر شماره ۳ آرایه computer به "keyboard" به کد اضافه کنید.

پاسخ :

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

کارگاه ۲ دریافت عناصر آرایه از کاربر

پاسخ به فعالیت‌ها

تکمیل کارگاه
ص ۸۶

- برنامه را تغییر دهید تا در ابتدا نام قطعه (عناصر آرایه computer) و سپس قیمت آن از ورودی دریافت شود .

پاسخ:

```
Computer[0] = Console.ReadLine();
Price[0] = int.Parse(Console.ReadLine());
```

- برنامه را برای دریافت نام و قیمت ۱۰ قطعه تغییر دهید.

پاسخ:

```
for (int i = 0; i <= 9; i++)
{
    computer[i] = Console.ReadLine();
    price[i] = int.Parse(Console.ReadLine());
}
```

- تعداد قطعات رایانه را از ورودی دریافت کنید.

پاسخ:

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

for (int i = 0; i <= count - 1; i++)
{
    computer[i] = Console.ReadLine();
    price[i] = long.Parse(Console.ReadLine());
}
```

کنجکاوی
ص ۸۶

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

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

```
13
2000
3000
4000
5000
System.Int64[]
```

پاسخ: خیر، در صورتی که از این دستور استفاده کنید برنامه خطای هنگام اجرا را نشان خواهد داد.

کارگاه ۳ نمایش عناصر آرایه

پاسخ به فعالیت‌ها

تکمیل کارگاه

ص ۸۹

- برنامه را با عدد ۷ اجرا کنید.

پاسخ:

چون عدد ۷ از تعداد داده‌های این آرایه بیشتر است بنابراین خطای هنگام اجرای برنامه رخ می‌دهد.

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

```
string[] week = { "sat", "sun", "mon", "tue", "wed", "thu", "fri" };
byte num = byte.Parse(Console.ReadLine());
Console.WriteLine(week[num]);

IndexOutOfRangeException was thrown
Index was outside the bounds of the array.
Troubleshooting tips:
- Make sure that indexes are correct.
- Make sure the index is not a negative number.
- Make sure that the maximum index for a list is less than the size.
Get general help for this exception.
```

پاسخ:

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

- دستوراتی به برنامه اضافه کنید که اگر شماره واردشده در محدوده مجاز نبود، پیام مناسب نمایش داده شود.

پاسخ:

```
if ((num < 1) || (num > 7))
    Console.Write("please enter correct number !!! ");
else
    Console.WriteLine(week[--num]);
```

تکمیل کارگاه

ص ۹۰

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

پاسخ:

```
for (byte i = 0; i <= 6; ++i)
{
    Console.WriteLine(week[i]);
    ++i;
}
```

کارگاه ۴ پیمایش آرایه

پاسخ به فعالیت‌ها

تکمیل کارگاه
ص ۹۰

- برنامه را طوری تکمیل کنید که نام، قیمت قطعات و قیمت کل را نمایش دهد.
پاسخ:

```
long[] price = new long[5];
string[] computer = new string[5];
long sumprice = 0;
for (int i = 0; i < 5; i++)
{
    computer[i] = Console.ReadLine();
    price[i] = long.Parse(Console.ReadLine());
}
for (int i = 0; i < price.Length; i++)
    sumprice += price[i];
for (int i = 0; i < 5; i++)
    Console.WriteLine(" {0} {1} ", computer[i],
price[i]);
Console.WriteLine(sumprice);
```

کارگاه ۵ پیمایش نویسه‌های رشته

پاسخ به فعالیت‌ها

تکمیل کارگاه
ص ۹۰

- دستورات لازم را برای دریافت یک رشته از کاربر بنویسید.
پاسخ:

```
string mystr;
mystr = Console.ReadLine();
- دستورات زیر را به کد اضافه کنید. خروجی آن چیست؟
for (int i = 0; i < mystr.Length; i++)
    Console.WriteLine(mystr[i]);
پاسخ: خروجی نمایش کاراکترهای رشته ورودی به صورت ستونی است.
- کد زیر را جایگزین مرحله ۳ کنید.
foreach (char c in mystr)
    Console.WriteLine(c);
پاسخ: دقیقاً همان رشته را تولید می‌کند.
- برای نمایش معکوس رشته، کد زیر را تکمیل کنید.
for (int i = mystr.Length - 1; i >= 0 ; i--)
```

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

- آیا می توان این کد را با دستور foreach نوشت؟

پاسخ : خیر

کنجکاوی

ص ۹۱

- در دستور foreach منظور از متغیر حلقه چیست ؟

پاسخ: متغیری است که مقادیر ذخیره شده در آرایه را در خود جای می دهد. به مثال زیر توجه کنید.

```
byte[] code = new byte[5];  
foreach (byte tmpcode in code)  
Console.WriteLine(tmpcode);
```

نوع داده آرایه از نوع byte تعریف شده است بنابراین نوع داده متغیر foreach هم از نوع byte تعریف شده است.

- آیا می توان به وسیله عملگر انتساب (=) مقداری را داخل متغیر حلقه foreach قرارداد ؟

پاسخ: خیر - تعریف و انتساب مقادیر متغیرها در دستور foreach امکان پذیر نیست.

فعالیت منزل

ص ۹۱

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

پاسخ:

```
byte[] shayes = new byte[10];  
byte count;  
  
for( byte i=0 ; i<=9 ; i++)  
{  
Console.Write("please enter number in 1,2,3 : " );  
shayes[i] = byte.Parse(Console.ReadLine());  
if ((shayes[i]<1) || (shayes[i]>3 ))  
{  
Console.WriteLine("please enter number in 1,2,3 !!!! " );  
shayes[i]=byte.Parse(Console.ReadLine());  
}  
}  
count=0;  
foreach(byte tmp in shayes)  
if(tmp == 1)  
count++;
```

```
Console.WriteLine(" tedad nomre shayesh ha-ye 1 ==> {0}
", count);
```

کارگاه ۶ ویرایش عناصر آرایه

پاسخ به فعالیت‌ها

تکمیل کارگاه
ص ۹۲

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

پاسخ: ابتدای کد اضافه شود:

```
byte i;
int temp;
int[] priceList = new int[50];
```

- دستورات محاسبه مالیات ارزش افزوده را بنویسید. در این کد نقش متغیر *i* و *z* چیست؟

پاسخ:

متغیر *i* قیمت محصولات وارد شده را نگهداری می‌کند.

متغیر *z* به تعداد محصولات وارد شده که در *i* ذخیره شده است امکان محاسبه مالیات بر ارزش افزوده و ذخیره در همان خانه متناظر را ایجاد می‌کند.

- در قطعه کد ۳ تبدیل صریح `int` را حذف کنید. چه خطایی رخ می‌دهد؟ چرا؟
پاسخ: چون محاسبه ارزش افزوده مقدار اعشاری (`double`) تولید می‌کند و فضای حافظه اشغالی توسط متغیر اعشاری بیشتر از فضای اشغالی متغیر صحیح (`int`) است بنابراین خطا رخ خواهد داد.



فعالیت منزل
ص ۹۲

برنامه‌ای 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.

پاسخ:

```
byte i;
byte z;
```

```

int temp;
int sum_cost;
int[] cost = new int[30];
for (i = 0; i < 30; i++)
{
    Console.WriteLine("Enter the cost of {0} th day : ", i +
1);
    temp = int.Parse(Console.ReadLine());
    if (temp > 0)
        cost[i] = temp;
    else
        break;
}
sum_cost = 0;
for (j = 0; j < i; j++)
    sum_cost = sum_cost + cost[j];
Console.WriteLine("sum cost of month ==> {0}",
sum_cost);
Console.WriteLine("average cost of day ==> {0} ",
((double)sum_cost / i));

```

شرح برنامه :

متغیر temp برای دریافت مقدار هزینه روزانه تعریف شده است، بنابراین اگر این هزینه صفر یا منفی باشد حلقه (i) ایجاد شده با دریافت چنین عددی خاتمه پیدا خواهد کرد و در حلقه دوم مجموع هزینه‌ها و میانگین آن‌ها، به تعداد روزها محاسبه خواهد شد (j) که در حلقه بالا برای آن‌ها هزینه روزانه دریافت شده است. بنابراین می‌بینید که متغیر j با متغیر i مقایسه شده است. در متغیر sum_cost مجموع هزینه‌ها حساب شده است. برای اینکه میانگین هزینه‌ها به صورت اعشاری نمایش داده شود از تبدیل مستقیم sum_cost / i (double) استفاده شده است.

جست‌وجو در آرایه

در این روش جستجوی اطلاعات معمولاً در فهرست نامرتب صورت می‌گیرد. در فهرست نامرتب، برای یافتن یک قلم اطلاعات خاص، تنها راه حل این است که عناصر فهرست را از اولین عنصر تا آخرین عنصر (یا بالعکس) با اطلاعات موردنظر مقایسه کنیم. اگر عنصری از فهرست، با اطلاعات موردنظر برابر باشد یا به پایان فهرست برسیم، عمل جستجو خاتمه می‌یابد.

فرض کنید فهرست زیر (آرایه X) موجود است :

X	X[0]	X[1]	X[2]	X[3]
	15	5	20	35

در این فهرست می‌خواهیم عدد ۲۰ را به طور خطی جستجو کنیم. مقدار ۲۰ را با خانه اول مقایسه می‌کنیم. اگر برابر نبود باید به خانه بعدی برویم و تا زمانی که محتوای خانه‌ای ۲۰ نباشد جستجو را ادامه دارد. این جستجو دو حالت دارد.

حالت اول: جستجوی موفق (یعنی مقدار جستجو شده در فهرست باشد)

حالت دوم: جستجوی ناموفق (یعنی مقدار جستجو شده در فهرست نباشد)

تعداد جستجوها وابسته به بودن و یا نبودن عنصر مورد نظر است. در صورتی که عنصر موجود باشد به شماره محل قرارگیری عنصر شمارش صورت گرفته است. (در خانه اول یک جستجو و در خانه دهم، ۱۰ جستجو) ولی اگر عنصر مورد نظر موجود نباشد، باید تمام خانه‌ها مورد جستجو قرار بگیرند که در این صورت تعداد جستجوها به اندازه طول آرایه خواهد بود.

پاسخ به فعالیت‌ها

فعالیت گروهی
ص ۹۳

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

پاسخ:

مکان عنصر مورد نظر	اول آرایه	وسط آرایه	آخر آرایه	در آرایه نباشد
تعداد مقایسه	۱	۵۰	۱۰۰	۱۰۰

فعالیت گروهی
ص ۹۴

در هریک از موارد زیر تعیین کنید از روش جستجوی خطی یا دودویی استفاده می‌شود؟ انتخاب روش بر اساس چه معیاری است؟

- یافتن کارنامه تحصیلی یک هنرجو در بین سایر کارنامه‌های مرتب‌شده بر اساس نام هنرجویان
- پیدا کردن یک کتاب بر اساس نام آن کتاب در کتابخانه
- پیدا کردن یک کتاب بر اساس کد کتاب در کتابخانه
- پیدا کردن نام هنرجویی که در کلاس بالاترین معدل را دارد، اگر اسامی بر اساس معدل مرتب شده باشند.

پاسخ:

ردیف	نوع جستجو	معیار انتخاب
۱	دودویی	فهرست کارنامه‌ها بر اساس نام هنرجویان مرتب شده است
۲	خطی	فهرست مرتب شده نیست
۳	خطی	فهرست مرتب شده نیست
۴	دودویی	فهرست هنرجویان بر اساس معدل آن‌ها مرتب شده است

کارگاه ۷ پیاده‌سازی جستجوی خطی

پاسخ به فعالیت‌ها

تکمیل کارگاه

ص ۹۴

- برای جست‌وجو در آرایه car کدهای زیر را بنویسید.
پاسخ:

این بخش برای آموزش جستجوی خطی بیان شده است. در روش جستجوی خطی مقدار مورد جستجو که در اینجا واژه prid است با محتوای تک‌تک خانه‌های آرایه مقایسه می‌شود. دو حالت رخ خواهد داد، در حالت اول محتوای خانه آرایه نام دیگری به‌غیر از prid است (مثلاً mazda) که در این حالت جستجو از خانه بعدی ادامه خواهد یافت و در حالت دوم مقدار خانه با واژه prid برابری خواهد کرد که در این صورت نام ماشین و محل خانه یافت شده به نمایش در خواهد آمد.

- برنامه را طوری تغییر دهید تا قیمت خودروی مورد جست‌وجو در خروجی نمایش دهنده شود.

پاسخ:

```
Console.WriteLine("{0} found in {1} and price {2}",  
item, i, price[i]);
```

- مقدار item را برابر Hyundai قرار دهید. خروجی چیست؟

پاسخ:

```
String item = " Hyundai ";
```

خروجی مانند قسمت قبل است با این تفاوت که به جای prid ماشین Hyundai جستجو و اطلاعات آن ماشین نمایش داده می‌شود.

- برای تشخیص یافتن و یا عدم یافتن عنصر، کد را به صورت زیر تغییر دهید. نقش متغیر found در این کد چیست؟

پاسخ:

متغیر found نقش یک پرچم را بازی می‌کند که در صورتی که مقدار آن true شود اطلاعات ماشین موردنظر نمایش داده می‌شود و در غیر این صورت یک پیام مناسب برای اطلاع کاربر نمایش می‌دهد.

- برنامه را طوری تغییر دهید که نام خودرو را از ورودی دریافت کند.

پاسخ:

```
item = Console.ReadLine();
```

فعالیت منزل

ص ۹۵

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.

پاسخ:

```
string[] nam = new string[10];  
double[] mark = new double[10];
```

```
byte i;
for (i = 0; i <= 9; i++)
{
    nam[i] = Console.ReadLine();
    mark[i] = double.Parse(Console.ReadLine());
}
for (i = 0; i <= 9; i++)
if (mark[i] == 20)
Console.WriteLine("mark {0} is 20 . ", nam[i]);
```

فعالیت گروهی

ص ۹۵

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 می‌کنید؟ بهتر است اولین حدس چه عددی باشد؟ پاسخ:

بهترین الگوریتم استفاده از عدد وسط هست (۵۰) حالا با توجه به بزرگ‌تر بودن یا کوچک‌تر بودن در نیمه اول اعداد (۱-۴۹) و نیمه دوم اعداد (۵۱-۱۰۰) باز هم باید حد وسط را ادامه دهیم.

هنرجویی عدد ۵۹ را یادداشت کرده است. در روش جست‌وجوی دودویی حداکثر با چند مقایسه عدد مورد نظر حدس زده خواهد شد. جدول زیر را کامل کنید. پاسخ:

بهتر است اولین حدس عدد ۵۰ باشد.

برای یافتن عدد ۵۹ با پنجمین مقایسه عدد به دست آمده است.

مقایسه	نتیجه مقایسه	عدد مورد نظر	عدد حدس زده شده (وسط آرایه)	آخرین عدد آرایه	اولین عدد آرایه
۱	عدد بزرگ‌تر است	۵۹	۵۰	۱۰۰	۱
۲	عدد کوچک‌تر است	۵۹	۷۵	۱۰۰	۵۱
۳	عدد کوچک‌تر است	۵۹	۶۲	۷۴	۵۱
۴	عدد کوچک‌تر است	۵۹	۵۶	۶۱	۵۱
۵	عدد درست است	۵۹	۵۹	۶۱	۵۷

- مقدار متغیر find پس از اجرای برنامه چیست؟ عملکرد متد IndexOf چیست؟

پاسخ:

مقدار find پس از اجرای برنامه ۲ است و عملکرد متد indexOf: اندیس اولین وقوع یک مقدار را در آرایه تعریف‌شده برمی‌گرداند.

```
variablename = Array.IndexOf(array_name, value);
```

- اگر $X=60$ قرار دهیم خروجی چه می‌شود؟

پاسخ:

به ازای مقدار $x=60$ برنامه اندیس ۱- را نمایش می‌دهد.

کد زیر را به برنامه اضافه کنید.

```
int find = Array.LastIndexOf(a, x);
```

```
Console.WriteLine(find);
```

مقدار متغیر find پس از اجرای این دستور چیست؟ تفاوت عملکرد متد LastIndexOf با متد IndexOf چیست؟

پاسخ:

مقدار find پس از اجرای برنامه ۶ است.

عملکرد متد LastIndexOf: اندیس آخرین وقوع یک مقدار را در آرایه تعریف‌شده برمی‌گرداند.

تفاوت متد IndexOf و LastIndexOf در این است که IndexOf اولین وقوع و LastIndexOf آخرین وقوع را برمی‌گرداند.

- کد زیر را به برنامه اضافه کنید.

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

پاسخ:

عملکرد متد sort(): عناصر موجود در آرایه تعریف‌شده را مرتب می‌کند.

- با دستور foreach عناصر آرایه a را در خروجی نمایش دهید. عملکرد متد sort چیست؟

پاسخ:

```
foreach (int temp in a)
```

```
Console.WriteLine(temp);
```

- کد زیر را به برنامه اضافه کرده، برنامه را اجرا کنید. مقدار find چیست؟

```
int x = 10;
```

```
int find = Array.BinarySearch(a, x);
```

```
Console.WriteLine(find);
```

پاسخ: مقدار find پس از اجرای برنامه 1 است .
 - کد زیر را به برنامه اضافه کنید و دوباره با دستور foreach عناصر آرایه a را در خروجی نمایش دهید.

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

پاسخ:

عملکرد متد Reverse(): عناصر موجود در آرایه تعریف شده را معکوس می کند.
 - کد زیر را اضافه کرده، برنامه را اجرا کنید. مقدار متغیر find چیست ؟

```
find = Array.BinarySearch(a, x);
```

```
Console.WriteLine(find);
```

پاسخ: مقدار find پس از اجرای برنامه 0 است .

تکمیل کارگاه
ص ۹۷

می خواهیم برنامه ای 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. گزینه های منو به رنگ سفید روی زمینه آبی نمایش داده شوند.

پاسخ:

```
byte[] myarray = new byte[5];
int find;
byte keysearch;

byte choose;
choose = 0;
do
{
    Console.BackgroundColor = ConsoleColor.Black;
    Console.Clear();
    Console.BackgroundColor = ConsoleColor.Blue;
    Console.ForegroundColor = ConsoleColor.White;
    Console.WriteLine("1-Input Array");
    Console.WriteLine();
    Console.WriteLine("2-Show Member Array");
    Console.WriteLine();
    Console.WriteLine("3-Sort Member Array");
    Console.WriteLine();
    Console.WriteLine("4-Reverse Member Array");
    Console.WriteLine();
    Console.WriteLine("5-Liner Search Into Array");
    Console.WriteLine();
    Console.WriteLine("6-Binary Search Into Array");
    Console.WriteLine();
    Console.WriteLine("7-Exit This Program");
    Console.WriteLine();
}
```

```

Console.BackgroundColor = ConsoleColor.Red;
Console.Write("Enter number (1-7) ==> ");
choose=byte.Parse(Console.ReadLine());
Console.BackgroundColor = ConsoleColor.Black;
switch(choose)
{
case 1 :
Console.WriteLine();
Console.WriteLine("=== enter 5 number for save into
array ==");
for (int i = 0; i < 5; i++)
myarray[i] = byte.Parse(Console.ReadLine());
break;
case 2 :
Console.WriteLine("== content your array ==");
for (int i = 0; i < myarray.Length; i++)
Console.WriteLine("myarray[{0}] ==> {1}", i,
myarray[i]);
Console.Write("press any key to continue...");
Console.ReadKey();
break ;
case 3 :
Array.Sort(myarray);
break;
case 4 :
Array.Reverse(myarray);
break;
case 5 :
Console.WriteLine("Enter number for search in your
array");
keysearch = byte.Parse(Console.ReadLine());
find = Array.IndexOf(myarray, keysearch);
if(find!=-1)
Console.WriteLine("Index your key in array ==> {0}" ,
find+1);
else
Console.WriteLine("your key is not found !!!!");
Console.ReadKey();
break;
case 6 :
keysearch = byte.Parse(Console.ReadLine());
Array.Sort(myarray);
find = Array.BinarySearch(myarray,keysearch);
if(find!=-6)
Console.WriteLine("Index your key in array ==> {0}",
find+1);

```

```

else
Console.WriteLine("your key is not found !!!!");
Console.ReadKey();
break;
case 7 :
Console.
WriteLine("Good By !!!!");
break;
default :
Console.WriteLine("please Enter correct number !!!!");
break;
}
} while (choose != 7);
Console.ReadKey();

```

کارگاه ۹ داده شمارشی رنگ‌ها

پاسخ به فعالیت‌ها

تکمیل کارگاه

ص ۱۰۰

- برنامه را اجرا کرده، خروجی را مشاهده کنید.

پاسخ:



- دستوری اضافه کنید که نام هر رنگ را به همان رنگ زمینه نمایش دهد.

پاسخ:

```

ConsoleColor c;
for (c = ConsoleColor.Black; c <= ConsoleColor.White;
c++)
{
    Console.BackgroundColor = c;
    Console.WriteLine(c);
}

```

کد زیر را به برنامه اضافه کنید.

```

c = 0;
Console.BackgroundColor = c + 3;

```

پاسخ:

```
Black
```

با استفاده از نوع داده شمارشی DayOfWeek برنامه‌ای بنویسید که اسامی روزهای هفته را همراه با شماره روز نمایش بدهد.

فعالیت منزل
ص ۱۰۰

پاسخ:

```
for (int i = 0; i <= 6; ++i)  
    Console.WriteLine(i + " : " + (DayOfWeek)i);
```

کارگاه ۱۰ داده شمارشی کلیدهای صفحه کلید

پاسخ به فعالیت‌ها

- کد زیر را در متد Main() بنویسید. برنامه را اجرا کرده، خروجی را مشاهده کنید.

تکمیل کارگاه
ص ۱۰۰

```
for (int i = 0; i < 128; i++)  
    Console.WriteLine(i + ":" + (ConsoleKey)i);
```

```
33:PageUp  
34:PageDown  
35:End  
36:Home  
37:LeftArrow  
38:UpArrow  
39:RightArrow  
40:DownArrow  
41:Select  
42:Print  
43:Execute  
44:PrintScreen  
45:Insert  
46>Delete  
47:Help  
48:D0  
49:D1  
50:D2
```

- کد را تغییر دهید به صورتی که برای نمایش خروجی از جا نگه‌دار استفاده کند.
پاسخ:

```
for (int i = 0; i < 128 ; i++)
    Console.WriteLine("{0,5}:{1}", i, (ConsoleKey)i);
```

فعالیت منزل
ص ۱۰۰

برنامه‌ای بنویسید که با دریافت هر کد، کلید معادل آن را نمایش بدهد. برنامه با ورود عدد صفر پایان یابد.
پاسخ:

```
byte x;
Console.Write("please enter code key ==>");
x = byte.Parse(Console.ReadLine());
while (x != 0)
{
    Console.WriteLine((ConsoleKey)x);
    x = byte.Parse(Console.ReadLine());
}
```

کنجکاوی
ص ۱۰۱

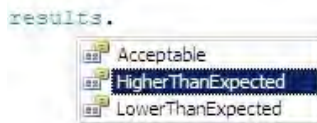
در نوع داده شمارشی Season نام Fall معادل با چه عددی است؟
پاسخ: در این نوع داده شمارشی Fall معادل عدد ۴ است.

کارگاه ۱۱ اعلان داده شمارشی

پاسخ به فعالیت‌ها

تکمیل کارگاه
ص ۱۰۲

- برنامه را اجرا کنید. کد و نتیجه را با مرحله ۳ مقایسه کنید.
پاسخ: با بررسی جواب هر دو قسمت مشاهده می‌کنیم که خروجی به‌دست‌آمده یکسان است. برای مثال اگر عدد ۲ را وارد کنید هر دو کد واژه Acceptable را نمایش خواهند داد. از آن جا که در داده شمارشی هر مقدار دارای یک شماره منحصر به فرد است و هر عدد در آن نوع داده متناظر با یک مقدار است، بنابراین در بخش اول عدد را با تعریف داده شمارشی مقایسه می‌کند و مقدار متناظرش را برمی‌گرداند و در بخش دوم از فن تبدیل نوع استفاده کرده‌ایم. یعنی مقدار متغیر را با تبدیل مستقیم به داده شمارشی متناظر آن تبدیل کرده‌ایم.
به مقادیر داده شمارشی تعریف‌شده نگاه بیندازید.



حالا دوباره به تعریف داده شمارشی نگاه کنید و تناظر این مقادیر با شماره‌های داده‌شده را به یاد آورید.

در کد بخش اول به سراغ داده‌های این داده شمارشی رفته است. (کد زیر)

```
case 1: Console.WriteLine(results.LowerThanExpected );
```

و در کد بخش دوم از روش تبدیل صریح برای تبدیل عدد تناظر با هر مقدار به مقدار ذخیره‌شده در داده شمارشی اقدام کرده است. (کد زیر)

```
Console.WriteLine((results)result);
```

- دستوراتی به برنامه اضافه کنید که اگر شماره واردشده در محدوده مجاز نباشد، پیام مناسب نمایش داده شود.

پاسخ:

```
if ((result >= 1) && (result <= 3) )
```

```
{
```

در این قسمت کدهای قسمت ۳ یا ۴ را یادداشت کنید

```
}
```

```
else
```

```
Console.WriteLine (" please enter number in 1 , 2 , 3 !!");
```

برنامه را تعمیم دهید به صورتی که نام هنرجویان یک کلاس و نمره یک واحد یادگیری آن‌ها را دریافت کرده، سپس سطح شایستگی همه هنرجویان کلاس را نمایش دهد.

پاسخ:

```
enum Results
```

```
{
```

```
    LowerThanExpected = 1,
```

```
    Acceptable = 2,
```

```
    HigherThanExpected = 3
```

```
}
```

```
static void Main(string[] args)
```

```
{
```

```
    int result;
```

```
    byte j = 1;
```

```
    byte k = 1;
```

```
    string[] name_stu = new string[3];
```

```
    byte[] mark_stu = new byte[3];
```

```
    for (byte i = 0; i < 3; ++i)
```

```
    {
```

```
        Console.Write("please enter name student {0}
```

```
==> ", j++);
```

```
        name_stu[i] = Console.ReadLine();
```

```
        Console.WriteLine("please enter mark (1/2/3)
```

فعالیت منزل

ص ۱۰۳

```

student {0} ==> ", k++);
    mark_stu[i] = byte.Parse(Console.ReadLine());
}
for (byte i = 0; i <3; ++i)
{
    Console.WriteLine("mark {0} is {1} ", name_stu[i],
mark_stu[i]);
    result = mark_stu[i];
    switch (result)
    {
        case 1:
            Console.WriteLine("student is not accept
!!!");
            Console.WriteLine("result is : {0} ",
(Results)result);
            break;
        case 2:
            Console.WriteLine("student is good accept
!!!");
            Console.WriteLine("result is : {0} ",
(Results)result);
            break;
        case 3:
            Console.WriteLine("student is well accept
!!!");
            Console.WriteLine("result is : {0} ",
(Results)result);
            break;
    }
}

```

فعالیت کارگاهی
ص ۱۰۳

کدام یک از دستورات مقداردهی قطعه کد زیر نادرست است؟ خطای آن‌ها چیست؟ آن‌ها را حذف کرده و بقیه را اجرا کنید.

```

DayOfWeek day;
day = Monday;
day = DayOfWeek.Wednesday;
Console.WriteLine(day);
day = 5;
day = (DayOfWeek)0;
Console.WriteLine(day);
Console.WriteLine((int)day);

```

پاسخ:

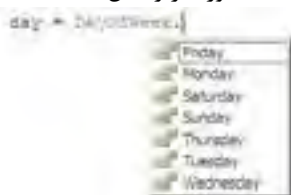

```

DayOfWeek day;
//      day = Monday;
day =
DayOfWeek.Wednesday;
Console.WriteLine(day);
//      day = 5;
day = (DayOfWeek)0;
Console.WriteLine(day);
Console.WriteLine((int)day);

```

خطوطی که بارنگ سبز مشخص شده‌اند دارای خطا هستند و باید حذف گردند.

خطای دستور `day = monday;` به علت دسترسی نادرست به مقدار داده تعریف شده است. برای چنین عملیاتی باید به صورت زیر عمل کنید.



خطای دستور `day = 5;` با هم به علت دسترسی نادرست به مقادیر ذخیره شده در کتابخانه داده شمارشی `DayOfWeek` است. یادمان باشد اگرچه هر داده تعریف شده در داده شمارشی دارای یک شماره منحصر بقرده است اما برای دسترسی به این مقادیر اجازه استفاده کردن از این شماره را نخواهیم داشت.

خطای دستور `day=(DayOfWeek)();` به علت استفاده نادرست از تبدیل صریح دستیابی به مقادیر داده‌ها است. برای استفاده صحیح از این تبدیل باید در پرانتز دوم مقدار عددی اختصاص یافته به یکی از داده‌های داده شمارشی `DayOfWeek` را وارد کنیم. مثلاً عدد ۵ که متناظر با `Friday` خواهد بود.

```

day = (DayOfWeek)(5);
Console.WriteLine(day);

```

برای مطالعه بیشتر

۱. آموزش گام به گام برنامه نویسی سی شارپ عین الله جعفر نژاد قمی ویراست ۵
2. <https://www.tutorialspoint.com/csharp/>
3. <http://csharp.net-tutorials.com/>
4. <https://www.microsoft.com/net/tutorials/csharp/getting-started>
5. <http://www.learncs.org/>
6. www.techulator.com/resources