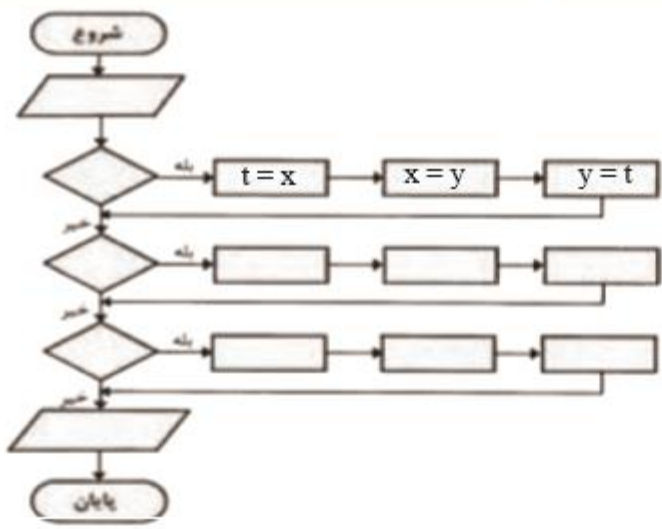


مسئله ۶: روندنمای طراحی شده را طوری تکمیل کنید که اندازه قد سه نفر را از ورودی گرفته، آنها را به صورت نزولی یعنی از بزرگ به کوچک مرتب کند (شکل ۴).

	ورودی‌ها	شناخت مسئله
	خروجی‌ها	
 <pre> graph TD Start([شروع]) --> Input[/ /] Input --> D1{ } D1 -- بله --> T1[t = x] T1 --> T2[x = y] T2 --> T3[y = t] D1 -- خیر --> D2{ } T3 --> D2 D2 -- بله --> T4[t = x] T4 --> T5[x = y] T5 --> T6[y = t] D2 -- خیر --> D3{ } T6 --> D3 D3 -- بله --> T7[t = y] T7 --> T8[y = x] T8 --> T9[x = t] D3 -- خیر --> Output[/ /] T9 --> Output Output --> End([پایان]) </pre>		روندنما



برای سه ورودی با ترتیب‌های مختلف روندنما را اجرا کرده، درستی آن را بررسی کنید.

مسئله ۷: هر رایانه رومیزی از موادی که در جدول رویه‌رو آمده، تشکیل شده است.
با فرض اینکه هر رایانه رومیزی حدود ۹ کیلوگرم وزن دارد، در کارگاه رایانه هترستان شما چندگرم طلا و نقره وجود دارد؟

ردیف	قطعه	درصد
۱	پلاستیک	۳۲
۲	سرب	۷
۳	آلومینیوم	۴۱
۴	طلا	۵/۶۱
۵	آهن	۵/۲
۶	نقره	۵/۹۸۱

شناخت مسئله	ورودی‌ها	تعداد رایانه‌های کارگاه
	خروجی‌ها	میزان طلا و نقره برحسب گرم
روندنما		

آشنایی با زبان سی شارپ

برنامه‌نویسی به زبان‌های سطح پایین و میانی به علت نزدیکی به زبان ماشین و سخت‌افزار، دشوارتر از زبان‌های سطح بالا است. زبان سی شارپ از زبان‌های سطح بالا است و به زبان محاوره‌ای نزدیک است.

12	-	$1V_{24} - 1V_1$	$2V_{24}$	$1V_{24} = 2V_{24}$	$1V_1 = 1V_1$	$= 24 - 21$
13	-	$1V_2 - 1V_1$	$2V_2$	$1V_2 = 2V_2$	$1V_1 = 1V_1$	$= 24 - 1$
14	+	$1V_1 + 1V_2$	$2V_2$	$1V_1 = 1V_1$	$1V_2 = 2V_2$	$= 2 + 1 = 3$
15	+	$2V_2 + 2V_2$	$1V_2$	$2V_2 = 2V_2$	$2V_2 = 2V_2$	$= 24$
16	=	$1V_2 = 2V_{24}$	$2V_{24}$	$1V_2 = 2V_2$	$1V_{24} = 2V_{24}$	$= 24 - 24$
17	-	$2V_2 - 1V_1$	$2V_2$	$2V_2 = 2V_2$	$1V_1 = 1V_1$	$= 24 - 2$
18	+	$1V_1 + 2V_2$	$2V_2$	$2V_2 = 2V_2$	$1V_1 = 1V_1$	$= 2 + 1 = 3$
19	=	$2V_2 = 2V_2$	$1V_2$	$2V_2 = 2V_2$	$2V_2 = 2V_2$	$= 24$
20	=	$1V_2 = 2V_{24}$	$2V_{24}$	$1V_2 = 2V_2$	$1V_{24} = 2V_{24}$	$= 24 - 24$
21	=	$1V_{24} = 2V_{24}$	$2V_{24}$	$1V_{24} = 2V_{24}$	$1V_{24} = 2V_{24}$	$= 24 - 24$

شکل ۶- اولین الگوریتم نوشته شده به وسیله نخستین برنامه‌نویس جهان خاتم ADA

زبان برنامه‌نویسی سی شارپ در سال ۲۰۰۰ با نام تجاری C# انتشار یافت. این زبان از خانواده زبان‌های برپایه .NET Framework است. زبان C# از لایه نرم‌افزاری .NET استفاده می‌کند که برای ویتدوز طراحی شده است. اگر بخواهیم روی سیستم عامل دیگری برنامه سی شارپ را اجرا کنیم، باید لایه نرم‌افزاری مطابق با .NET، روی آن سیستم نصب شود.

آشنایی با ویژوال استودیو

IDE (Integrated Development Environment) محیط‌های نرم‌افزاری توسعه یافته و یکپارچه است. IDE‌هایی که برای برنامه‌نویسی استفاده می‌شوند، نرم‌افزارهایی هستند که ابزارهای مورد نیاز مانند ویرایشگر برنامه، مترجم، اشکال زدایی و ... را به صورت یکپارچه برای برنامه‌نویس فراهم می‌سازند. در این کتاب از نسخه رایگان Visual Studio Express 2012 for Desktop استفاده می‌کنیم که محیط IDE برنامه‌نویسی مایکروسافت است و آن را به اختصار VS می‌نامیم.

مراحل اجرای یک پروژه جدید در سی شارپ:

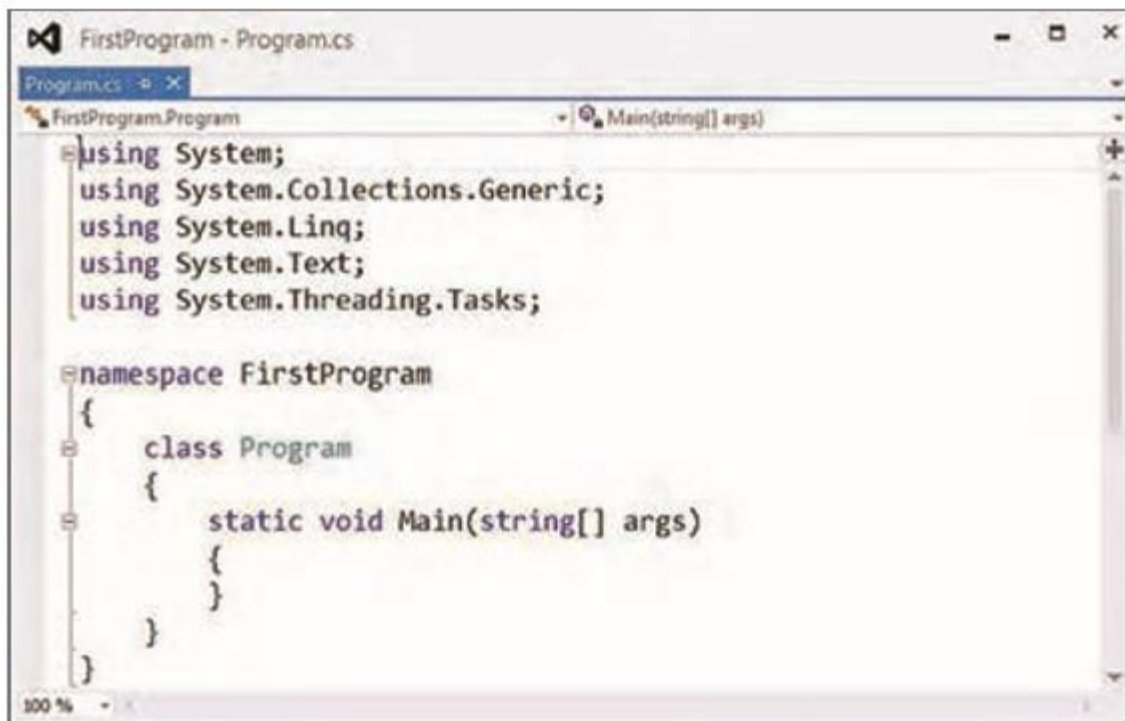
- ۱- برنامه را اجرا کنید
- ۲- New project را کلیک کنید
- ۳- سمت چپ گزینه c# را انتخاب کنید
- ۴- سمت راست گزینه console application را انتخاب کنید.

۵- وارد محیط کد نویسی برنامه می شوید .



شکل ۷- ایجاد پروژه

در این کتاب برنامه‌نویسی در مد **Console Application** انجام می‌شود. برنامه‌های نوشته شده در این مد، **برنامه کنسول** نامیده می‌شوند. در برنامه کنسول امکان استفاده از واسطه‌های گرافیکی (GUI) وجود ندارد. ورودی‌ها و خروجی‌های برنامه کنسول در پنجره خط فرمان (Command Prompt) قرار می‌گیرند.

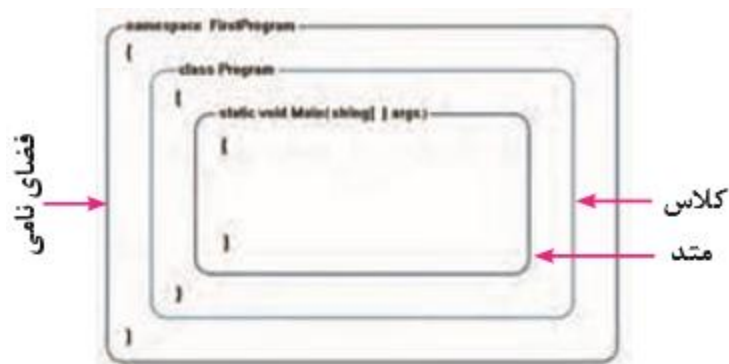


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace FirstProgram
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

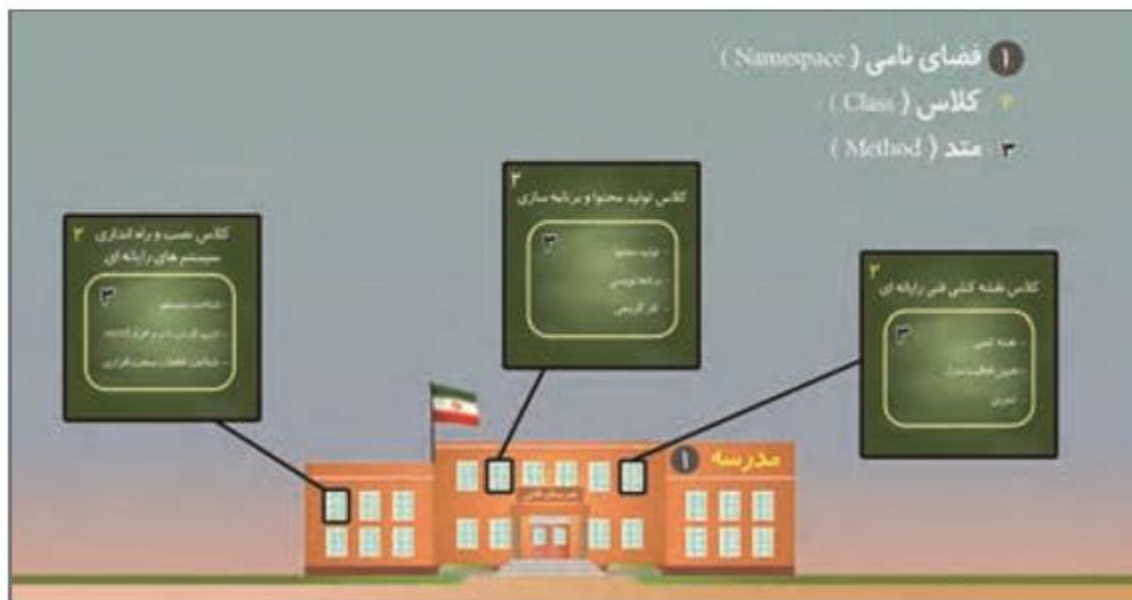
شکل ۹- یک برنامه ساده درسی شارپ

شکل ۹ الگوی یک برنامه ساده درسی شارپ را نشان می‌دهد. در این الگو `using`, `namespace`, `class`, `static`, `void` و `string` کلید واژه‌های سی شارپ هستند و با رنگ یکسان مشخص می‌شوند. البته نگران نباشید، با ایجاد یک پروژه، سی شارپ الگو را در اختیار برنامه‌نویس قرار می‌دهد. بنابراین نیازی به حفظ کردن آنها ندارید.



شکل ۱۰- الگوی یک برنامه ساده سی شارپ

مطابق الگوی شکل ۱۰ متد **Main** که نقطه شروع برنامه‌های سی‌شارپ است، در داخل کلاس قرار دارد. به صورت پیش فرض نام فضای نامی، با نام پروژه یکسان است. فضای نامی مشابه فضای مدرسه است که شامل چندین کلاس است، کلاس‌های مدرسه مشابه کلاس در سی‌شارپ است و فعالیت‌های درون کلاس همان متدها هستند.



شکل ۱۱- نمونه‌ای از فضای نامی کلاس و متد در دنیای واقعی

۳ در پنجره ویرایشگر برنامه دستور زیر را وارد کنید.

`Console.WriteLine("in the name of god");`

در این دستور از متد **WriteLine** برای نمایش پیام در صفحه خروجی استفاده شده است. این متد در کلاس **کنسول (Console)** قرار دارد. کلاس **کنسول** از کلاس‌های آماده سی‌شارپ است.

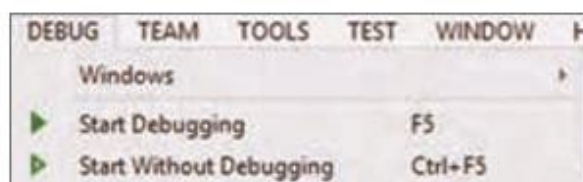
```
in the name of god
Press any key to continue . . .
```

۴ با استفاده از کلید **Ctrl+ F5** برنامه را اجرا کرده، خروجی را مشاهده کنید.

۵ بار دیگر برنامه را با کلید **F5** اجرا کنید.

پنجره کنسول بلافاصله بسته می‌شود؛ برای توقف برنامه و مشاهده پنجره خروجی، دستور: **Console.ReadKey()** را در انتهای برنامه اضافه کرده، برنامه را اجرا کنید.

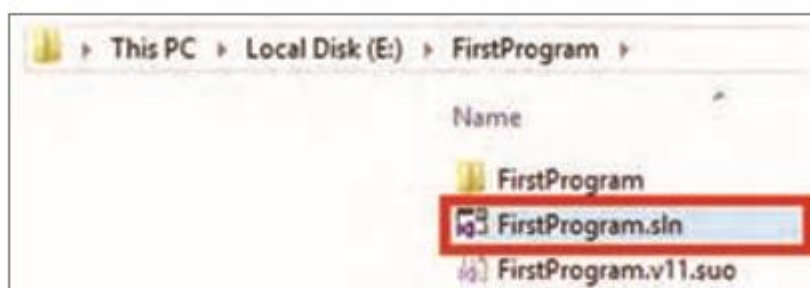
```
static void Main(string[] args)
{
    Console.WriteLine("in the name of god");
    Console.ReadKey();
}
```



شکل ۱۲- منوی Debug

متد **ReadKey()** برای دریافت کلیدی از صفحه کلید به کار می‌رود. این متد سبب توقف برنامه تا فشردن یک کلید می‌شود و به کاربر اجازه مشاهده خروجی را می‌دهد. این متد در کلاس کنسول قرار دارد.

برای بازکردن مجدد پروژه می‌توانید در پوشه پروژه روی فایل با پسوند **.sln** دابل کلیک کنید (شکل ۱۳).



شکل ۱۳- پوشه پروژه

نکته ۱ : در انتهای تمامی دستورات سی شارپ از علامت **(:)** نقطه ویرگول استفاده می شود .

نکته ۲ :

زبان سی شارپ به حروف کوچک و بزرگ حساس (Case sensitive) است.

یادداشت



۱ پروژه جدیدی با نام Screen ایجاد کنید.

۲ کد زیر را در Main وارد کنید.

```
Console.Write("I ");
Console.Write("like ");
Console.Write("C#");
```

۳ برنامه را اجرا کرده، خروجی آن را یادداشت کنید.

۴ کد قبلی را پاک کرده، کد زیر را جایگزین کنید.

```
Console.WriteLine("I ");
Console.WriteLine("like ");
Console.WriteLine("C#");
```

برنامه را اجرا کرده، خروجی آن را با کد قبلی مقایسه کنید.

۵ برنامه را پاک کرده، قطعه کد زیر را وارد کنید.

برای نوشتن دستور Console.WriteLine، می‌توانید cw را تایپ کرده و سپس کلید tab را دوبار فشار دهید. برنامه را اجرا کرده، نتیجه را با مرحله ۴ مقایسه کنید.

```
Console.WriteLine("I ");
Console.WriteLine();
Console.WriteLine("like ");
Console.WriteLine();
Console.WriteLine("C#");
```

۶ در قطعه کد مرحله ۵ دستور WriteLine را به Write تبدیل کنید.

۷ برنامه را اجرا کرده، نتیجه را با مرحله ۵ مقایسه کنید.

نکته ۳ :

دستور write () را نمی‌توان بدون پارامتر ورودی به کار برد.

یادداشت



از این پتجره برای نوشتن کدهای برنامه استفاده می‌شود (شکل ۱۶). این محیط دارای ویژگی‌هایی برای سرعت بخشیدن به فرایند برنامه‌نویسی است. باز شدن خودکار فهرست هوشمند (Intellisense) که برای نوشتن صحیح و سریع‌تر مورد استفاده قرار می‌گیرد، از قابلیت‌های این محیط است. برای باز شدن این فهرست می‌توان از کلید ترکیبی **Ctrl+Space** استفاده کرد.



شکل ۱۶- فهرست هوشمند

کارگاه ۴ ایجاد صدا با فرکانس‌های مختلف

- متد **Beep** برای ایجاد یک صدا در برنامه استفاده می‌شود. شکل کلی این دستور به صورت زیر است:
- Console.Beep (مدت زمان برحسب میلی ثانیه، فرکانس برحسب هرتز)**
- ۱ پروژه جدیدی با نام **Melody** ایجاد کنید.
 - ۲ برنامه زیر را در **Main** وارد کنید.

```
Console.WriteLine("I");
Console.Beep(200, 800);
Console.WriteLine("like");
Console.Beep(200, 800);
Console.WriteLine("C#");
Console.Beep(200, 800);
```

۳ برنامه را اجرا کنید.

۴ اعداد درون دستور `Console.Beep(,)` را به دلخواه تغییر دهید و برنامه را اجرا کنید.

۵ اعداد درون متد را حذف کنید و برنامه را اجرا کنید.

خروجی با مرحله قبل چه تفاوتی دارد؟

توجه داشته باشید مقدار فرکانس را باید در محدوده متناسبی بنویسید زیرا گوش انسان تنها قادر است اصواتی با فرکانس حدود ۲۰۰ تا ۱۰۰۰۰ هرتز را بشنود. صداهای بم فرکانس کم و صداهای زیر فرکانس بالا دارند.

با کم و زیاد کردن اعداد ورودی‌های متد `Beep` روشی را برای زیر و بم کردن صدا پیدا کنید.

فعالیت
کارگاهی



قطعه برنامه زیر را با فرکانس و مدت زمان‌های داده شده کامل کنید و از شنیدن موسیقی که نوشته‌اید، لذت ببرید. برای ادامه موسیقی از هترآموز خود کمک بگیرید.

فعالیت
کارگاهی



```
Console.Beep(370, 600);
```

```
Console.Beep(550, 600);
```

```
Console.Beep(370, 600);
```

1	2	3	4	5
(550, 600)	(590, 300)	(590, 600)	(490, 300)	(590,600)
(550,300)	(550,1200)	(550, 600)	(660,300)	(550,600)
(590, 300)	(700, 600)	(490, 300)	(550,1200)	(490,600)
(660, 300)	(660, 600)	(440, 300)	(370,600)	(440,300)

در این برنامه می‌خواهیم متن‌های محیط کنسول را با رنگ‌های مختلف چاپ کنیم.

۱ پروژه جدیدی با نام `color1` ایجاد کنید.

۲ کد زیر را در `Main` وارد کنید. و برنامه را اجرا کنید.

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

```
Console.WriteLine(" ");
```

```
Console.ForegroundColor = ConsoleColor.Red;
```

```
Console.WriteLine(" ");
```

۳ نام و نام خانوادگی خود را به ترتیب در دستور `WriteLine` اول و دوم قرار دهید و دوباره برنامه را اجرا کنید.

تفاوت خروجی مرحله ۲ و ۳ چیست؟

۴ برنامه را به شیوه زیر تغییر داده، دوباره اجرا کنید. تفاوت آن با مرحله قبل چیست؟

```
Console.BackgroundColor = ConsoleColor.Blue;
```

```
Console.Clear();
```

```
Console.ForegroundColor = ConsoleColor.Red;
```

```
Console.WriteLine("ali");
```

۵ در انتهای کد مرحله ۴ دستور `Console.Clear()` را بنویسید و برنامه را اجرا کنید تفاوت آن با مرحله قبل چیست؟

۶ با توجه به دستورات جدیدی که به کار گرفتید، جدول زیر را تکمیل کنید.

ردیف	دستور	کاربرد دستور
۱	<code>Console.ForegroundColor</code>	رنگ قلم نوشته
۲	<code>ConsoleColor.Blue</code>	
۳	<code>Console.BackgroundColor</code>	
۴	<code>Console.Clear();</code>	



- به کمک هم گروهی خود برنامه‌ای بنویسید که نام شما را با رنگ قرمز روی زمینه آبی نمایش دهد و پس از شنیدن صدای بوق به مدت یک ثانیه رنگ صفحه نمایش را به قرمز تغییر داده، نام هم گروهی شما را روی آن به رنگ آبی نمایش دهد.

- برنامه را تغییر دهید که به جای مکث یک ثانیه با فشار کلیدی از صفحه کلید، نام هم گروهی شما را نمایش دهد. از متد `ReadKey` در بین کدهای برنامه استفاده کنید.



`ConsoleColor` جعبه رنگ ۱۶ تایی در `C#` است.

یادداشت



برنامه‌ای بنویسید که جدولی از همه رنگ‌های سی‌شارپ به همراه نام رنگ در کنسول نمایش دهد.

فعالیت منزل



۱۸۵

با استفاده از دستور `SetCursorPosition(left, top)` با دادن شماره ستون و سطر (فاصله از چپ و بالا) می‌توان مکان نما را به محل دلخواه در پنجره کنسول منتقل کرد.

یادداشت



۱ پروژه جدید با نام ConsoleFlag ایجاد کنید.

۲ قطعه کد زیر را در متد Main وارد کنید.

```
Console.BackgroundColor = ConsoleColor.Green;
Console.SetCursorPosition(29, 9);
Console.WriteLine(" ");

Console.SetCursorPosition(29, 10);
Console.WriteLine(" ");

Console.BackgroundColor = ConsoleColor.White;
Console.SetCursorPosition(29, 11);
Console.WriteLine(" ");

Console.SetCursorPosition(29, 12);
Console.WriteLine(" ");
```

۱۸۶

فصل چهارم: حل مسائل ساده



شکل ۱۸

۳ برنامه را اجرا کنید.

۴ از برنامه الگو بگیرید و برنامه‌ای بنویسید که شکل ۱۸ را تولید کند.

۵ در ابتدای خطی که زمینه را سفید می‌کند، نویسه‌های // قرار داده، برنامه را اجرا کنید. خروجی را با خروجی مرحله ۵ مقایسه کنید.

۶ در ابتدای یک خط برنامه نویسه‌های * / قرار داده، در انتهای چند خط پایین‌تر نویسه‌های * / را قرار دهید و برنامه را اجرا کرده، خروجی را با مرحله ۵ مقایسه کنید.



برنامه‌ای بنویسید که پله‌های رنگی مطابق شکل زیر را ترسیم کند.

فعالیت منزل



📌 پروژه جدیدی با نام ConsoleDraw ایجاد کنید.

۲ قطعه کد زیر را در Main وارد کنید.

برای ترسیم از جدول زیر استفاده کنید به این ترتیب که پس از انتخاب کد مناسب با استفاده از کلید **Alt**، کد مورد نظر را وارد کنید از شماره‌های، بخش ماشین حساب صفحه کلید باید استفاده شود.

کد	علامت	کد	علامت	کد	علامت
Alt+219	■	Alt+222	▮	Alt+176	░░░░░░░░░░
Alt+220	■	Alt+223	■	Alt+177	░░░░░░░░░░
Alt+221	▮	Alt+248	°	Alt+178	░░░░░░░░░░

[illegible]

۲ برنامه را اجرا کنید.
۱ با استفاده از دستورات رنگ، تصویر را رنگ آمیزی کنید.

موفق باشید

میرحسینی