



به نام خدا
دانشکده‌ی مهندسی برق و کامپیوتر دانشکده فنی
دانشگاه تهران
مبانی کامپیوتر و برنامه‌نویسی



استاد : دکتر مرادی، دکتر هاشمی

دستور کار آزمایشگاه شماره 2
آشنایی مقدماتی با C، متغیرها و کار با
ورودی و خروجی

نیمسال اول 01-02

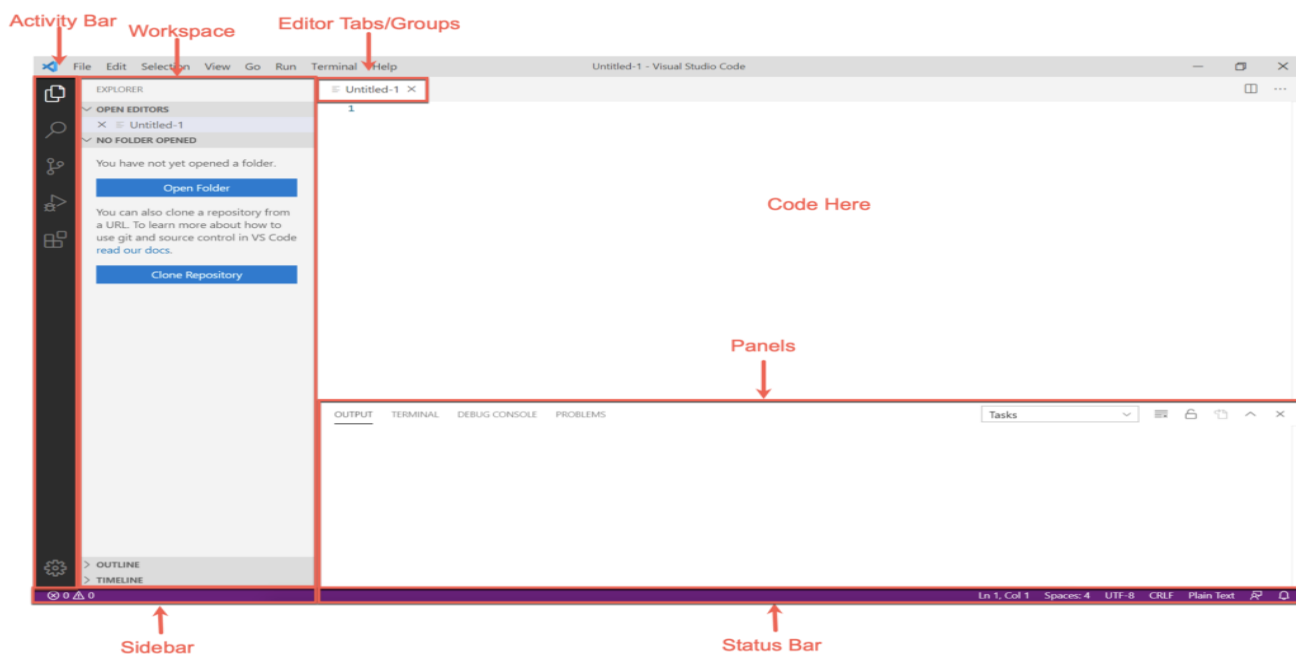
در این جلسه شما قرار است تمرین بیش تری در محیط برنامه نویسی VSCode 1.52 بکنید و با نحوه ی استفاده از توابع ورودی و خروجی بیش تر آشنا شوید.

پیش نیاز:

مطالعه و انجام خودآموز VSCode 1.52

هر گاوچرانی به یک اسب و هر کدنویس به یک ویرایشگر کد عالی نیاز دارد. که ما در این آموزش قرار است VSCode 1.52 را آموزش دهیم...

وقتی برای اولین بار این برنامه را اجرا می کنید یک صفحه مانند شکل پایین مشاهده خواهید کرد که اجزای آن را به مرور توضیح خواهیم داد.



Editor Window:

پنجره ویرایشگر جایی است که شما بیشتر کار خود را انجام خواهید داد. این صفحه جایی است که می توانید تمام کدهایی را که روی آن کار می کنید مشاهده و ویرایش کنید. هر زمان که یک فایل جدید را باز می کنید یا یک فایل موجود را ویرایش می کنید، پنجره ویرایشگر جایی است که کد نمایش داده می شود. VS Code دارای برگه هایی در این ویرایشگر است که به شما امکان می دهد چندین فایل را همزمان باز کنید و گروه های ویرایشگر که برگه های مختلف را گروه بندی می کنند.

Workspace:

رایج ترین بخش بعدی رابط کاربری^۱ خواهد بود که استفاده می کنید. فضای کاری^۲ جایی است که هر فایلی که در برگه ها^۳ باز کرده اید و همچنان پوشه ای که در آن قرار دارید را نشان می دهد. از این قسمت می توانید دسترسی راحت تری به سایر فایل های پروژه داشته باشید.

Panels:

بخش پانل ها بخش "خروجی" است. در این بخش «برگه های» مختلفی را با اطلاعاتی که توسط VS Code و پسوند های آن برگردانده شده اند پیدا خواهید کرد. اینجا جایی است که ترمینال را نیز پیدا خواهید کرد.

Status Bar:

نوار وضعیت: در این قسمت می توانید اطلاعات مربوط به پروژه و وضعیت آن مانند تعداد خطاها و هشدار ها و همچنین نوع فایل و ... را مشاهده کنید.

Extensions

یکی از مهم ترین ویژگی های وی اس کد، افزونه های آن است. VS Code از ویژگی های رایج ویرایشگر کد مانند برجسته سازی دستور زبان، Intellisense و غیره پشتیبانی می کند. نوع کدی که می نویسید را درک می کند و به طرق مختلف با آن سازگار می شود. اما همه زبان ها را نمی فهمد. C فقط یک متن ساده است. برای اینکه VS Code بدانند چه زمانی یک بخش باید جمع شود و چه متنی باید رنگ را تغییر دهد، باید ساختار C را درک کند. VS Code بسته به نوع فایل باز شده، این ویژگی ها را متفاوت اعمال می کند.

¹ user interface

² Work space

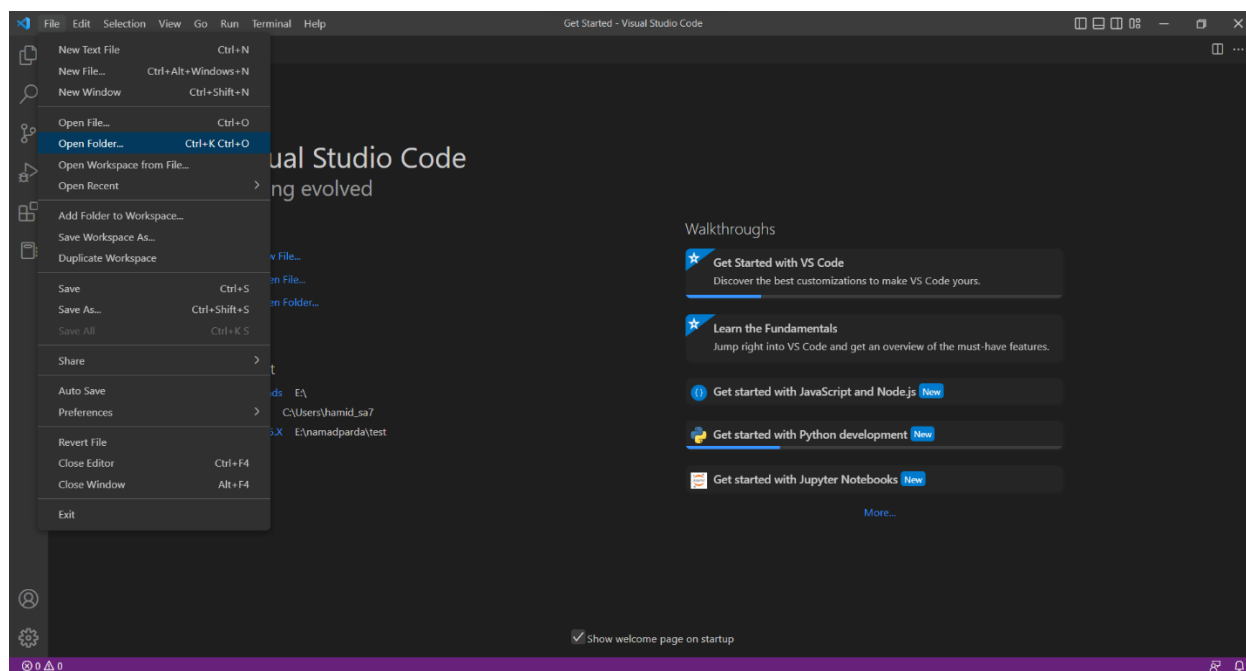
³ tabs

همانطور که در سر کلاس گفته شد برنامه به زبان C احتیاج به کامپایل شدن دارد. برای این کار نیاز داریم تا کامپایلر استاندارد gcc را روی سیستم نصب نمایید. برای این کار از برنامه ی مدیریت کامپایلر minGW استفاده می کنیم.

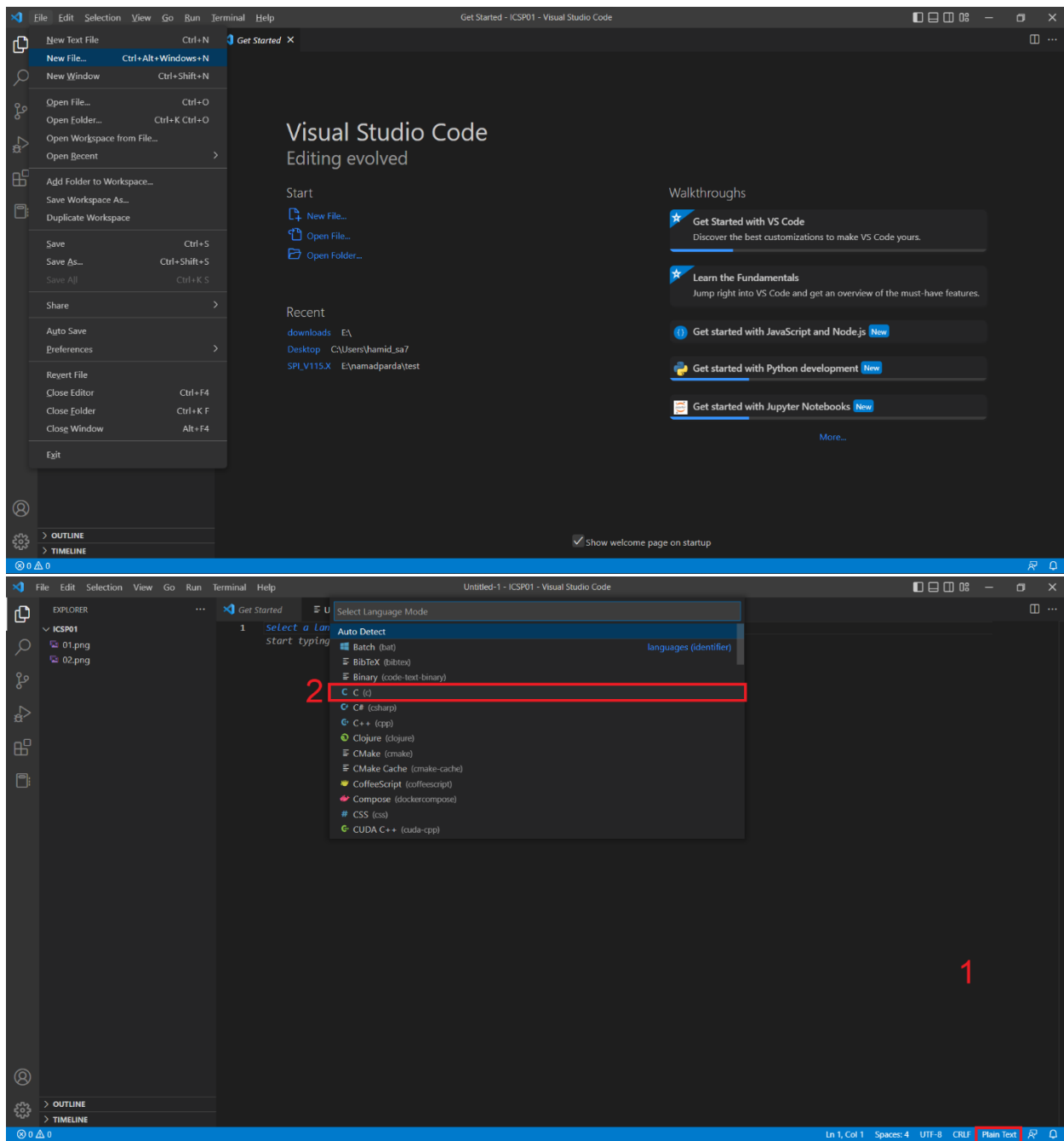
طبق دستورات این [لینک](#) این کامپایلر را نصب نمایید.

1. ← انجام دهید! (یک برنامه ی ساده)

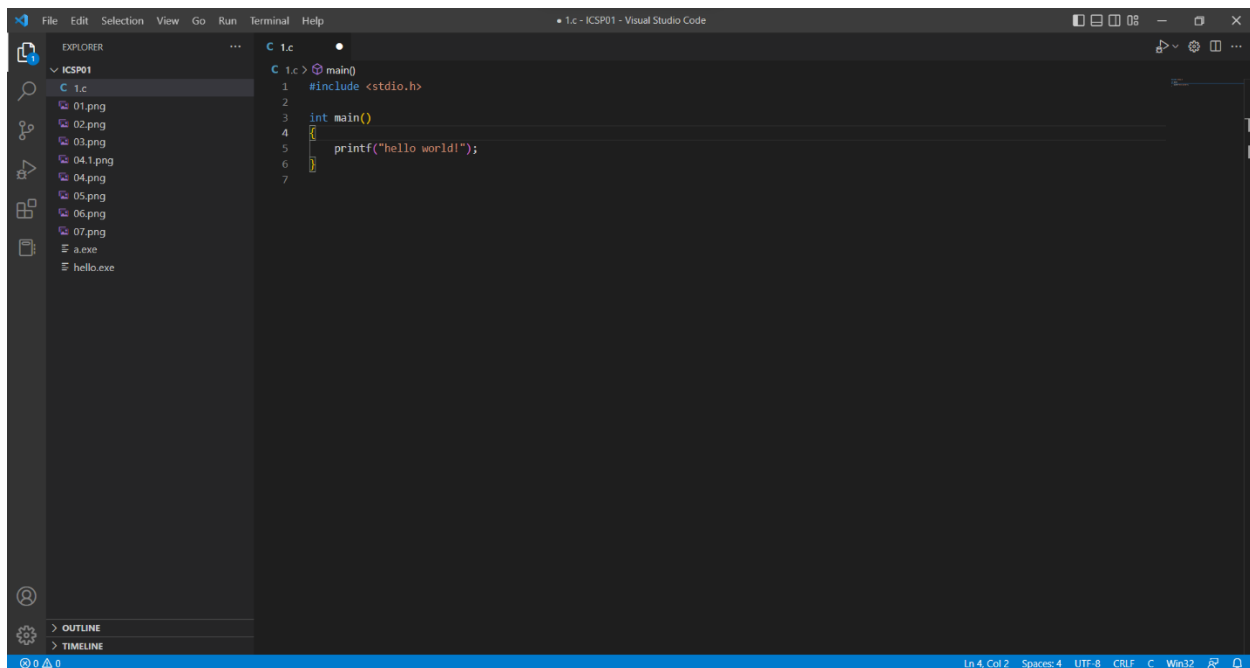
ابتدا در VS code پوشه ای که که برای ذخیره ی پروژه مد نظر دارید را مطابق تصویر باز کنید.



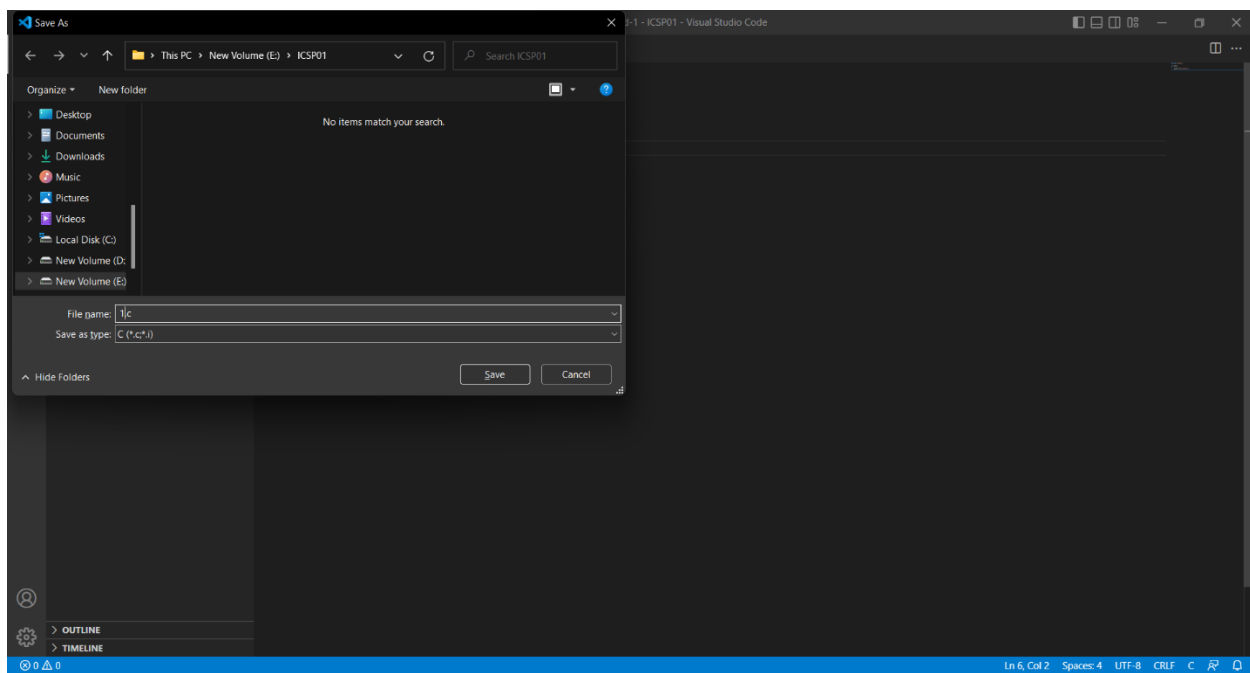
حال یک فایل متنی را مطابق تصویر اضافه کرده و به فایل زبان C تغییر می دهیم.



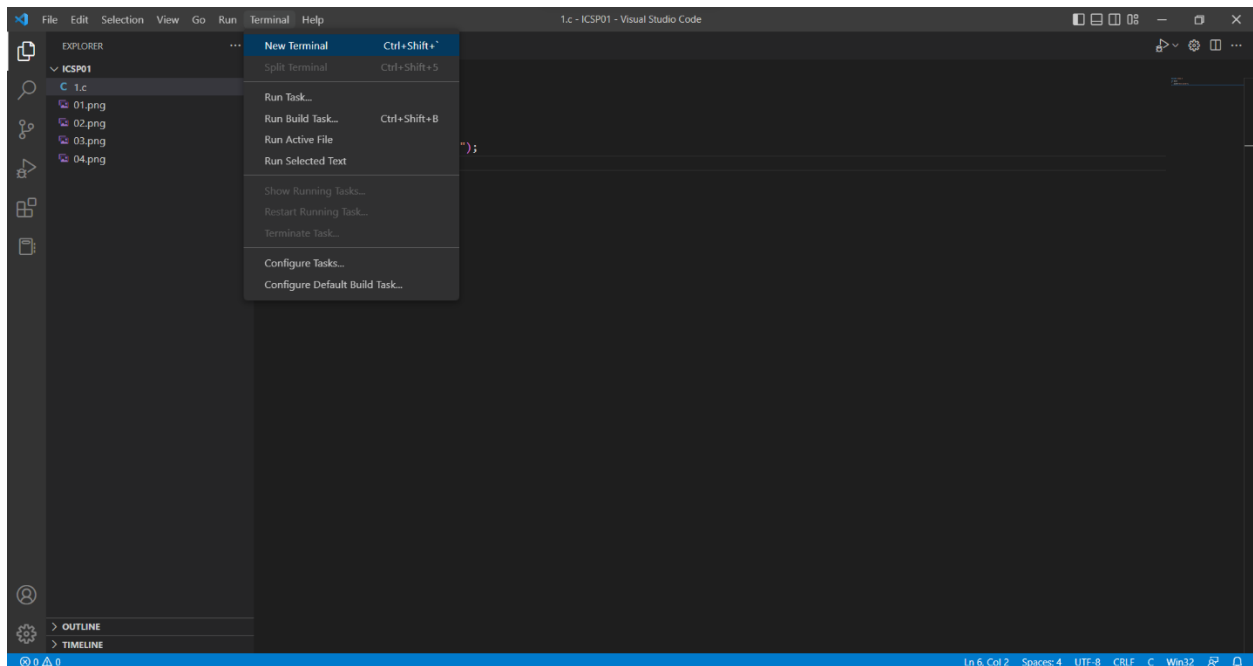
سپس کد را می نویسم:



و آن را با نام دلخواه (در اینجا 1) ذخیره می‌کنیم.



سپس ترمینال را با کلیدهای ترکیبی **Ctrl + Shift + `** و با مطابق تصویر زیر باز می‌کنیم.



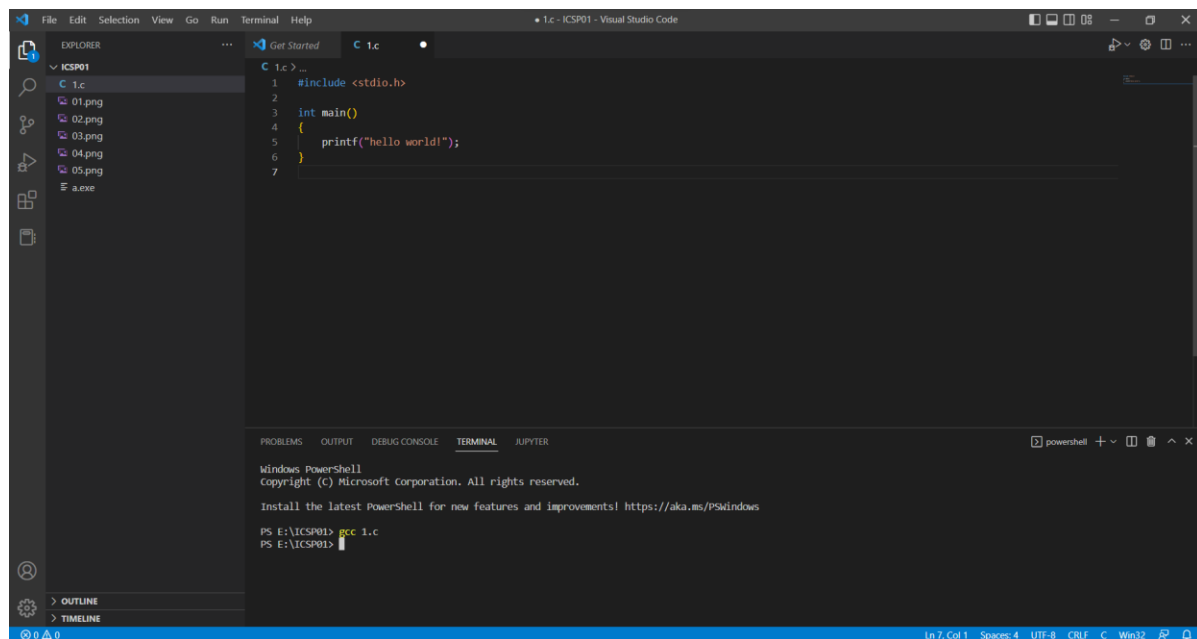
حال با وارد کردن دستور

gcc 1.c

فایل مورد نظر را کامپایل می‌کنیم. همانطور که در پنجره‌ی کناری و فولدر می‌بینید فایل برنامه (a.exe) اضافه شده است.

همچنین می‌توانیم با تغییر دستور به حالت زیر نام فایل را تغییر دهیم:

gcc 1.c -o hello.exe

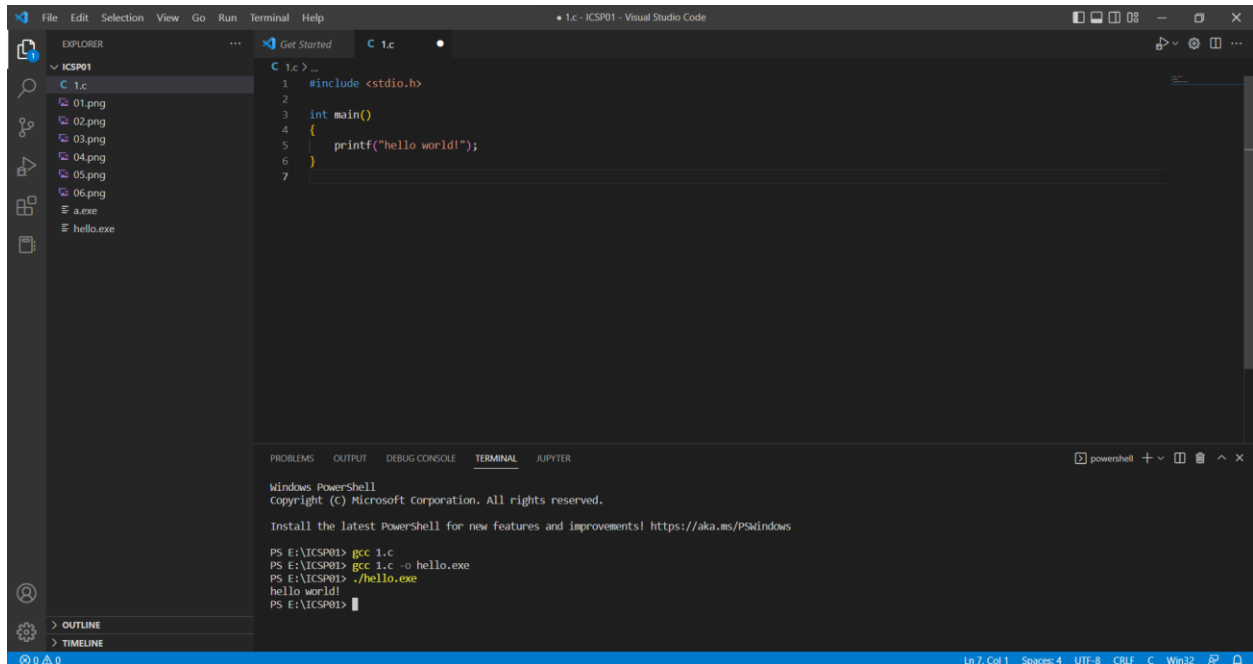


سپس با وارد کردن نام فایل :

./hello.exe

برنامه‌ی نوشته شده را اجرا می‌کنیم.

0



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left shows a project named 'ICSP01' with files '01.png' through '06.png', 'a.exe', and 'hello.exe'. The Editor pane shows a C file named '1.c' with the following code:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("hello world!");
6 }
7
```

The Terminal pane at the bottom shows the execution of the program in a Windows PowerShell window:

```
PS E:\ICSP01> gcc 1.c
PS E:\ICSP01> gcc 1.c -o hello.exe
PS E:\ICSP01> ./hello.exe
hello world!
PS E:\ICSP01>
```

همانطور که در تصویر می‌بینید، عبارت **hello world!** را نمایش می‌دهد.

قسمت 1 : نتایج را به دستیاران آموزشی نشان دهید.

عملیات خواندن از ورودی و نوشتن در خروجی توسط دو تابع **Printf** و **Scanf** :

توابع **scanf** و **printf** به ترتیب توابع ورودی و خروجی استاندارد فرمت دار هستند. یعنی شما می‌توانید فرمت داده ای که می‌خواهید بخوانید و یا بنویسید را تعیین کنید.

1. فرمت های مختلف در جدول زیر آمده است:

Format	Format Specifier
int	%d or %i
char	%c
float	%f
double	%lf
string	%s

فرمت های دیگری نیز وجود دارند. مثلاً **%x** برای اعداد **hex** و ...

2. انجام دهید! (یک برنامه ی ساده ی دیگر) ←

1. حال این قطعه کد را در یک پروژه ی جدید اجرا کنید تا توضیحات بالا بیش تر برایتان جا بیفتد.

```
#include <stdio.h>
int main() {
    int x, y;
    scanf("%d", &x);
    scanf("%d", &y);
    printf("The result is: %d\n", ((x + y) << 2) % 3);
}
```

2. سعی کنید قطعه کد بالا را جوری تغییر دهید که کاربر بتواند دو عدد ورودی را در یک خط و با یک کاراکتر space بین آن دو وارد کند.

می توانید برای پیدا کردن پاسخ سوال خود به لینک زیر مراجعه کنید:

<https://stackoverflow.com/questions/1412513/getting-multiple-values-with-scanf>

قسمت 2 : حال نتایج را به دستیاران آموزشی نشان دهید.

3. در تابع scanf، \n در انتهای فرمت قرار ندهید. اگر قرار دهید کاربر باید یک enter اضافه وارد کند.

4. در دستور scanf، علامت & که قبل از x گذاشته شده را بردارید و مجدداً برنامه را کامپایل و اجرا نمایید. حال یک عدد را به عنوان ورودی وارد کنید. چه اتفاقی افتاد؟ چرا برنامه از کار افتاد؟ در این مورد به دو نکته ی زیر توجه کنید:

1. شما علت دقیق بروز این خطا را بعد از آشنایی با مفاهیم تابع⁴ و نحوه ی آرگومان دهی⁵ به آن، اشاره گر ها⁶ و مرجع ها⁷ در همین درس خواهید آموخت، ولی در حال حاضر همین قدر بدانید که اگر در تابع scanf علامت & قبل از x را فراموش کنید بگذارید، برنامه ی شما می خواهد به خانه ای از حافظه دسترسی داشته باشد که مال خودش نیست! برای همین سیستم عامل جلوی اجرای برنامه را می گیرد.

2. خطایی که با آن مواجه شدید خطایی بود که به هنگام اجرا رخ داد⁸ و خطای زمان کامپایل⁹ نبود. (زیرا برنامه شما به درستی و بدون خطا کامپایل شد).

⁴ Function

⁵ Argument Passing

⁶ Pointers

⁷ References

⁸ Runtime Error

⁹ Compile Error

5. خط اول برنامه فوق (یعنی `<stdio.h>` `#include`) را حذف کنید و مجدداً برنامه را کامپایل نمایید. چه اتفاقی می افتد؟
پیغام خطایی که کامپایلر به شما می دهد به چه معنا است؟

قسمت 3 : علت را برای دستیاران آموزشی توضیح دهید.

← 3. انجام دهید!

متغیرها در کامپیوتر به روش های مختلفی ذخیره می شوند. از این روش ها می توان روش ASCII برای متغیر از نوع char و یا سیستم نمایش Floating Point را برای اعداد اعشاری یا متغیر float نام برد.

1. قطعه کد زیر را در یک پروژه جدید اجرا کنید. سپس به عنوان ورودی کاراکتر S را وارد نمایید.

```
#include <stdio.h>
int main() {
    char x;
    printf("Enter a character:\n");
    scanf("%c", &x);
    printf("%d\n", x);
}
```

قسمت 4 : عدد مشاهده شده در خروجی نمایانگر چه مقداری است؟ علت را برای دستیاران آموزشی توضیح دهید.

2. قطعه کد زیر را در یک پروژه جدید اجرا کنید. سپس به عنوان ورودی یک بار عدد 5 و یک بار عدد 1092091904 را وارد کنید.

```
#include <stdio.h>
int main() {
    float x;
    printf("Enter a decimal number:\n");
    scanf("%d", &x);
    printf("%f\n", x);
}
```

عدد مشاهده شده در خروجی نمایانگر چه مقداری است؟ مقدار عدد مشاهده شده در خروجی به ازای ورودی 1092091904 را در سیستم Floating Point محاسبه کنید. سپس با استفاده از مبدل اعداد binary به decimal در لینک زیر مقدار عدد را در مبنای 10 محاسبه کنید. چه نتیجه ای می گیرید؟

<http://www.binaryhexconverter.com/binary-to-decimal-converter>

قسمت 5 : نتیجه را برای دستیاران آموزشی توضیح دهید.

4. انجام دهید!

1. برنامه ای بنویسید که عدد اعشاری r را از کاربر بگیرد و مساحت دایره ای به شعاع r را حساب کند و نتیجه را تا 3 رقم اعشار نمایش دهد. مثلاً برای حالت $r = 10$ خروجی به صورت زیر نمایش است:

The result is: 314.160

2. راهنمایی 1:

```
#include <stdio.h>
#define PI 3.1416

int main() {
    double r, result;
    //...
}
```

3. راهنمایی 2: برای نمایش یک عدد اعشاری تا 3 رقم اعشار می توانید از یکی دیگر از قابلیت های تابع `printf` استفاده کنید! به کد زیر توجه کنید:

```
float r;
printf("%.3f\n", r);
```

با این کار می توانید عدد اعشاری r را تا 3 رقم اعشار چاپ کنید.

فرمت ها در دستور `printf` بسیار پرکاربرد هستند و می توانند خروجی شما را خوانا و زیبا کنند. برای آشنایی بیش تر با فرمت ها می توانید به لینک زیر مراجعه کنید:

<http://www.codingunit.com/printf-format-specifiers-format-conversions-and-formatted-output>

4. توجه 1: یکی از ویژگی های یک برنامه ی خوب کاربر پسند¹⁰ بودن آن است. برای رعایت این نکته پیش از خواندن ورودی ابتدا باید به کاربر پیغام مناسب بدهید. (مانند قطعه کد صفحه ی قبل)

5. توجه 2: به عبارت `#define PI 3.1416` که در ابتدای قطعه کد بالا نوشته شده است توجه کنید با این تعریف در هر جای کدتان می توانید به جای عدد 3.1416 از label ای به نام PI استفاده کنید!

قسمت 6 : حال نتایج را به دستیاران آموزشی نشان دهید.

5. انجام دهید!

ابتدا یک متغیر از جنس `float` تعریف کنید و مقدار اولیه آن را نیز 0.3 قرار دهید. اکنون قطعه کد زیر را اجرا کنید:

```
#include <stdio.h>
```

¹⁰ User friendly

```
int main() {
    float x = 0.3;
    printf("%.30f\n", x + x + x + x + x + x + x + x + x + x);
}
```

حال تعداد اعشار را به 5 تا تغییر دهید. چه مشاهده می‌کنید؟

قسمت 7: علت نتیجه بدست آمده را به دستیاران آموزشی توضیح دهید.

← 6. انجام دهید! (امتیازی)

1. قطعه کد زیر را در یک پروژه جدید اجرا کنید. سپس به عنوان ورودی عدد 115 را وارد کنید.

```
#include <stdio.h>
int main() {
    char x;
    printf("Enter a number:\n");
    scanf("%d", &x);
    printf("%c\n", x);
}
```

مشاهده می‌کنید که برنامه پس از اجرا با خطای runtime مواجه می‌شود. علت این خطا چیست؟

قسمت 8: نتیجه را با دستیاران آموزشی مطرح کنید.

موفق باشید