



به نام خدا

دانشکده‌ی مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران
مبانی کامپیوتر و برنامه‌نویسی



استاد : دکتر مرادی

دستور کار آزمایشگاه شماره ۲
آشنایی مقدماتی با C، متغیرها و کار
با ورودی و خروجی

نیمسال دوم ۹۸-۹۹

در این جلسه شما قرار است تمرین بیش تری در محیط برنامه نویسی Visual Studio 2015 انجام دهید و با نحوه ی استفاده از توابع ورودی و خروجی بیش تر آشنا شوید.

پیش نیاز:

مطالعه و انجام خودآموز Visual Studio 2015.

۱. انجام دهید! (یک برنامه ی ساده) ←

۱. یک پروژه ی جدید بسازید و برنامه ای بنویسید که خروجی زیر را چاپ کند:

Hello World!

۲. راهنمایی:

```
#include <stdio.h>
int main() {
    printf("Hello World!\n");
}
```

قسمت ۱: نتایج را در کادر زیر بنویسید .

با توجه به برنامه داده شده در راهنمایی، نوشته Hello World! پرینت خواهد شد.

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 می توانید فرمت داده ای که می خواهید بخوانید و یا بنویسید را تعیین کنید.

۱. 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 ...

← ۲. انجام دهید! (یک برنامه ی ساده ی دیگر)

۱. 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);
}
```

۲. 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.

قسمت ۲: نتایج را در کادر زیر بنویسید.

کد خواسته شده بصورت زیر خواهد بود:

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

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

۴. 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؟

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

قسمت ۳: علت موارد فوق را در کادر زیر بنویسید .

۳- علامت `\n` به معنای new line می باشد و نوشتن آن داخل " " ، کاربر را مجبور میکند تا فرمت خواسته شده در `scanf` را کامل بزند که یعنی باید یک `Enter` اضافه وارد نماید.

۴- برنامه ارور میدهد که متغیر `initialize` یا مقداردهی اولیه نشده است.

۵- با انجام این کار برنامه ارور میدهد که توابع `scanf` و `printf` حل نشده اند و برنامه، آنها را نمی شناسد.

← ۳. انجام دهید!

متغیرها در کامپیوتر به روش‌های مختلفی ذخیره می‌شوند. از این روش‌ها می‌توان روش ASCII برای متغیر از نوع char و یا سیستم نمایش Floating Point را برای اعداد اعشاری یا متغیر float نام برد.

۱. 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 ۴: عدد مشاهده شده در خروجی نمایانگر چه مقداری است؟ علت را نیز در کادر زیر بنویسید.

عدد 115 که عدد ASCII حرف s میباشد نمایش داده میشود.

۲. قطعه کد زیر را در یک پروژه جدید اجرا کنید. سپس به عنوان ورودی یک بار عدد ۵ و یک بار عدد ۱۰۹۲۰۹۱۹۰۴ 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 ۱۰۹۲۰۹۱۹۰۴ را در سیستم Floating Point محاسبه کنید. سپس با استفاده از مبدل اعداد binary به decimal در لینک زیر مقدار عدد را در مبنای ۱۰ محاسبه کنید. چه نتیجه‌ای می‌گیرید؟

قسمت ۵: نتیجه را در کادر زیر بنویسید.

خروجی عدد 5 برابر 0.000000 و خروجی عدد 1092091904 برابر 9.500000 میباشد.
میتوان نتیجه گرفت که اگر مقدار خروجی طبق استاندارد IEEE به عدد باینری و سپس به دسیمال تبدیل تبدیل شود ، اگه عدد ورودی را به ما میدهد.

۴. انجام دهید! 

۱. برنامه ای بنویسید که عدد اعشاری ۲ را از کاربر بگیرد و مساحت دایره ای به شعاع ۲ را حساب کند و نتیجه را تا ۳ رقم اعشار نمایش دهد. مثلاً برای حالت $r = 10$ خروجی به صورت زیر نمایش است:

The result is: 314.160

۲. راهنمایی ۱:

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

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

۳. راهنمایی ۲: برای نمایش یک عدد اعشاری تا ۳ 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 ۲ را تا ۳ رقم اعشار چاپ کنید.

قابل توجه است که فرمت ها در دستور `printf` بسیار پرکاربرد هستند و می توانند خروجی شما را خوانا و زیبا کنند.

۴. توجه ۱: یکی از ویژگی های یک برنامه ی خوب کاربر پسند^۱ بودن آن است. برای رعایت این نکته پیش از خواندن ورودی ابتدا باید به کاربر پیام مناسب بدهید. (مانند قطعه کد صفحه ی قبل)

۵. توجه ۲: 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 ۳,۱۴۱۶ از label ای به نام `PI` استفاده کنید!

قسمت ۶: حال نتایج بدست آمده را در کادر زیر بنویسید.

```
#include <stdio.h>  
#define PI 3.1416  
int main() {  
    double r;  
    scanf("%lf", &r);  
    printf("the result is: %.3f\n", PI*r*r);  
}
```

 **۵. انجام دهید!**

ابتدا یک متغیر از جنس `float` تعریف کنید و مقدار اولیه آن را نیز ۰,۳ قرار دهید. اکنون قطعه کد زیر را اجرا کنید:

¹ User friendly

```
#include <stdio.h>
int main() {
    float x = 0.3;
    printf("%.30f\n", x + x + x + x + x + x + x + x + x + x);
}
```

حال تعداد اعشار را به ۵ تا تغییر دهید. چه مشاهده می‌کنید؟

قسمت ۷: علت نتیجه بدست آمده را در کادر زیر بنویسید.

ابتدا عدد 2.99999976158142089843750000000000 نمایش داده میشود که علت خطا این است که ابتدا سیستم عدد رو به عدد باینری تبدیل میکند و این کار رو با تقریب انجام میدهد. ولی وقتی تعداد اعشار را به 5 تا تغییر میدهیم ، برنامه متناسب با آن، عدد را گرد میکند و عدد 3.00000 را به ما نشان میدهد. خب نتیجه این است که فرمت float دقت لازم رو برای کارهای بسیار دقیق ندارد. و گاها با گرد کردن عدد همراه است. پس میتوان از آن در جاهایی که گرد کردن عدد اهمیتی ندارد استفاده کرد.

← ۶. انجام دهید! (امتیازی)

۱. قطعه کد زیر را در یک پروژه جدید اجرا کنید. سپس به عنوان ورودی عدد ۱۱۵ 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 ۸: نتیجه را در کادر زیر بنویسید.

ما ابتدا x را کاراکتر معرفی میکنیم ولی در scanf از کاربر میخواهیم که یک عدد دسیمال وارد کند و ما متغیر آن را دوباره x معرفی میکنیم به همین علت برنامه دچار خطای runtime میشود.

موفق باشید