به نام خدا

دانشکده‌ی مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران

مبانی کامپیوتر و برنامه‌نویسی

نیمسال دوم 98-99	دستور کار آزمایشگاه شماره 2 آشنایی مقدماتی با C، متغیرها و کار با ورودی و خروجی	استاد : دکتر مرادی

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.

1. انجام دهید! (یک برنامه ی ساده)

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

```
Hello World!
```

2. راهنمایی:

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

قسمت 1 : نتایج را در کادر زیر بنویسید .

کدی که در راهنمایی آمده است عبارت hello world! را در console چاپ می کند

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

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.

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.

قسمت 2 : نتایج را در کادر زیر بنویسید.

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

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؟

5. خط اول برنامه فوق (یعنی `#include <stdio.h>`) 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 موارد فوق را در کادر زیر بنویسید .

3. اگر \n در انتهای scanf قرار بگیرد به جای 2 ورودی 3 ورودی دریافت می شود دلیل این است که scanf بر اساس white space کار می کند یعنی ورودی می گیرد تا به کاراکتر غیر white space برسد در اینجا چون \n به صورت white space است 2 ورودی را می گیرد و بعد از دریافت \n منتظر می ماند تا یک غیر white space دریافت کند تا دریافت ورودی به پایان برسد

4. در صورن نبودن & آدرس متغیر ما به scanf برای تغییر مقدار آن داده نمی شود و برنامه به آدرسی که برای خودش نیست می خواهد دسترسی پیدا کند که جلوی آن گرفته می شود برنامه به درستی کامپایل می شود ولی در هنگام اجرا به ارور می خوریم

5. عبارت `#include <stdio.h>` برای دریافت ورودی و چاپ کردن خروجی است stdio.h شامل توابع scanf و printf است و بدون این عبارت این دو تابع بی معنی خواهند بود و برنامه خطای کامپایل خواهد داشت

3. 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.

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 نیز در کادر زیر بنویسید.

عدد نمایش داده شده عدد ascii کاراکتر وارد شده است به عنوان مثال عدد ascii کاراکتر a عدد 97 است که در صورت وارد کردن a در برنامه فوق عدد 97 به ما داده می شود و به ازای s عدد 115 نمایش داده می شود

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 ار مبدل اعداد binary به decimal در لینک زیر مقدار عدد را در مبنای 10 محاسبه کنید. چه نتیجه‌ای می گیرید؟

قسمت 5 : نتیجه را در کادر زیر بنویسید.

عدد خروجی 9.500000 است که در سیستم floating point برابر 01000001000110000000000000000000 است که اگر این عدد را به decimal تبدیل کنیم برابر عدد ورودی می شود پس روند برنامه به این شکل است که ابتدا عدد ورودی را به binary تبدیل می کنیم سپس عدد حاصل را در سیستم floating point به decimal تبدیل می کنیم عدد حاصل خروجی خواهد بود

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 ونتیجه را تا 3 رقم اعشار نمایش دهد. مثالاً 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.

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

4. توجه 1: یکی از ویژگی های یک برنامه ی خوب کاربر پسند[1] 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 بدست آمده را در کادر زیر بنویسید.

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

int main() {
	double r, result;
	printf("Please Enter Your Number For Radius:\n");
	scanf("%lf", &r);
	result = PI * r * r;
	printf("Area of the circle: %.3f\n", result);
}
```

5. انجام دهید!

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

[1] User friendly

```c
#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 5 تا تغییر دهید. چه مشاهده می‌کنید؟

قسمت 7: علت نتیجه بدست آمده را در کادر زیر بنویسید.

با 30 رقم اعشار عدد به طور دقیق برابر 3 نمی شود ولی وقتی با 5 رقم اعشار نمایش می دهیم عدد به صورت دقیق برابر 3 می شود دلیل این است که اعدادی که به صورت float هستند دقیق نمی باشند و به صورت حدودی می باشند به خاطر این که در سیستم floating point اعداد اعشاری مثل 0.3 به صورت حدودی به دست می آیند و دقیق برابر 0.3 نمی شوند پس وقتی که با 30 رقم اعشار دقت نمایش عدد را خیلی بالاتر می بریم به دلیل این که عدد به صورت حدودی است دیگر 0.3 نمایش داده نمی شود ولی وقتی تا 5 رقم اعشار نمایش می دهیم چون دقت خیلی بالا نیست عدد به صورت دقیق نمایش داده می شود یعنی در نهایت عدد 2.99999 به 3 رند می شود

6. انجام دهید! (امتیازی)

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

```c
#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 در کادر زیر بنویسید.

علت این خطا این است که &x آدرس قسمتی از حافظه را می دهد که یک کاراکتر در آن قرار می گیرد و یک بیت را اشغال کرده است و می توان به میزان یک بیت در آن ذخیره کرد ولی در scanf وقتی که %d را قرار می دهیم scanf یک عبارت integer که 4 بیت است را از ورودی دریافت می کند و آن را در x قرار می دهد و خطا در اینجا رخ می دهد زیرا ورودی 4 بیتی را می خواهیم در متغیر 1 بیتی ذخیره کنیم و چون به اندازه کافی جا نداریم بر روی stack بقیه اطلاعات overwrite می شود و با خطا runtime مواجه می شویم

موفق باشید