



به نام خدا

دستور کار کارگاه مبانی کامپیوتر و برنامه‌نویسی

جلسه پنجم



محاسبات در C – ورودی و خروجی‌ها در C

محاسبات در C

۱. برنامه‌ای بنویسید که با دریافت دو ضلع قائم یک مثلث قائم الزاویه از کاربر، که از جنس double می‌باشند وتر این مثلث را محاسبه نماید. نتیجه را نیز به صورت یک عدد double نمایش دهید.

ورودی:

3 4

خروجی:

5.0

۲. برنامه‌ای بنویسید که با دریافت چهار نقطه به صورت (x, y) با فرض اینکه هیچ سه راسی روی یک خط قرار ندارند، طول اضلاع چهار ضلعی حاصل را گزارش کند.

ورودی:

0 0

0 1

1 1

1 0

خروجی:

1.000000

1.000000

1.000000

1.000000

۳. یک ثابت با نام PI تعریف کنید و آن را با عدد زیر مقداردهی کنید:

3.14159265359

حال با دریافت شعاع به صورت یک عدد double از کاربر محیط و مساحت دایره را به صورت double محاسبه کرده و چاپ کنید.

ورودی:

4

خروجی:

```
P = 25.132741
S = 50.265482
```

۴. یک متغیر از جنس `int` تعریف کرده و با `scanf` مقدار آن را از کاربر بخوانید. در `scanf` به اشتباه از `%d` استفاده نکرده و از `%lf` استفاده کنید. چه اتفاقی می‌افتد؟

۵. متغیری را از جنس `int` تعریف کرده و آن را با `۱۰،۱` مقداردهی کنید. آیا `cast` رخ می‌دهد؟ در این صورت این `cast` از نوع `explicit` می‌باشد یا `implicit`؟

ورودی و خروجی‌ها در C

در ادامه قصد داریم با نحوه‌ی کار با ورودی و خروجی‌ها در زبان C بیشتر آشنا شویم. همی برنامه‌ها نیاز دارند که با کاربر خود در ارتباط باشند که این ارتباط از طریق ورودی‌ها و خروجی‌ها صورت می‌گیرد. این ارتباط باید به گونه‌ای باشد که کاربر به سادگی و آنطور که نیاز دارد با سیستم شما ارتباط برقرار کند و در این صورت است که شما یک برنامه موفق نوشته‌اید.

آنچه برای مشخص کردن فرمت در `printf` استفاده می‌کنید از ساختار زیر پیروی می‌کند:

```
%[flags][width][.precision][length]specifier
```

برای اطلاعات بیشتر می‌توانید از [اینجا](#) استفاده کنید.

flags	description
-	Left-justify within the given field width; Right justification is the default (see width sub-specifier).
+	Forces to precede the result with a plus or minus sign (+ or -) even for positive numbers. By default, only negative numbers are preceded with a - sign.
(space)	If no sign is going to be written, a blank space is inserted before the value.
#	Used with o, x or X specifiers the value is preceded with 0, 0x or 0X respectively for values different than zero.

width	description
(number)	Minimum number of characters to be printed. If the value to be printed is shorter than this number, the result is padded with blank spaces. The value is not truncated even if the result is larger.
*	The width is not specified in the format string, but as an additional integer value argument preceding the argument that has to be formatted.

.precision	description
.number	For integer specifiers (d, i, o, u, x, X): precision specifies the minimum number of digits to be written. If the value to be written is shorter than this number, the result is padded with leading zeros. The value is not truncated even if the result is longer. A precision of 0 means that no character is written for the value 0. For a, A, e, E, f and F specifiers: this is the number of digits to be printed after the decimal point (by default, this is 6). For g and G specifiers: This is the maximum number of significant digits to be printed. For s: this is the maximum number of characters to be printed. By default all characters are printed until the ending null character is encountered. If the period is specified without an explicit value for precision, 0 is assumed.
.*	The precision is not specified in the format string, but as an additional integer value argument preceding the argument that has to be formatted.

۶. فرض کنید از شما خواسته شده است که مساحت یک قطعه زمین مستطیل شکل را محاسبه کنید. ورودی‌ها طول و عرض زمین می‌باشند که به متر هستند. این برنامه برای اینکه بتواند پاسخگوی نیازهای مختلف افراد باشد ورودی‌ها را به صورت اعداد حقیقی از نوع double در نظر می‌گیرد.

از آنجایی که ورودی‌های این برنامه به صورت اعداد حقیقی از نوع double می‌باشند، خروجی‌های آن نیز اعداد حقیقی از نوع double خواهند بود. راه‌های مختلفی برای چاپ خروجی حقیقی در C وجود دارد، بخشی از آن‌ها به شرح زیر می‌باشد. آن‌ها را امتحان کرده و نتیجه‌ی هر یک را با مدرس خود بررسی نمایید. برای بهتر دیده شدن نتایج از طول برابر ۲ و عرض برابر ۱ استفاده کنید.

```
printf("%lf\n", area);
printf("%.2lf\n", area);
printf("%.02lf\n", area);
printf("%04.2lf\n", area);
printf("%05.2lf\n", area);
printf("%g\n", area);
```

در صورتی که به جای %lf از %f استفاده شود چه اتفاقی رخ خواهد داد؟

آنچه برای مشخص کردن فرمت در scanf استفاده می‌کنید از ساختار زیر پیروی می‌کند:

```
%[*][width][length]specifier
```

برای اطلاعات بیشتر می‌توانید از [اینجا](#) استفاده کنید.

sub-specifier	description
*	An optional starting asterisk indicates that the data is to be read from the stream but ignored (i.e. it is not stored in the location pointed by an argument).
width	Specifies the maximum number of characters to be read in the current reading operation (optional).
length	One of hh, h, l, ll, j, z, t, L (optional).

۷. می‌خواهیم برنامه‌ای بنویسیم که تاریخ را از کاربر گرفته و آن را به تفکیک روز، ماه و سال چاپ کند.

ورودی:

```
14/11/1372
```

خروجی:

```
year = 1372 month = 11 day = 14
```

در برنامه فوق کاربر به اشتباه از کاراکتر – استفاده می‌کند، متن زیر برگرفته از سایت cplusplus.com می‌باشد که در آن رفتار scanf در شرایطی که کاربر برخلاف کاراکترهای شما ورودی وارد می‌کند را توضیح می‌دهد.

Any character that is not either a whitespace character (blank, newline or tab) or part of a format specifier (which begin with a % character) causes the function to read the next character from the stream, compare it to this non-whitespace character and if it matches, it is discarded and the function continues with the next

character of format. If the character does not match, the function fails, returning and leaving subsequent characters of the stream unread.

۸. تفاوت بین دستورات زیر در چه موردی می‌باشد؟

```
char c1 = getchar();
```

```
char c2;  
scanf("%c", &c2);
```

به نظر شما از این دستورات در چه مواردی می‌توان استفاده کرد؟

۹. یک شرکت تولیدی گاو صندوق برنامه‌ای می‌خواهد که از طریق آن بتواند برای گاو صندوق خود رمز تولید کند. این برنامه به این ترتیب می‌باشد که کاربر در ابتدا یک عدد به عنوان تعداد حداکثر ارقام رمز وارد می‌کند. در ادامه برنامه یک عدد تصادفی تولید می‌کند که محدودیت حداکثر ارقام تولید شده را دارا می‌باشد به این معنا که اگر رمز حداکثر ۳ رقمی است عدد تولید شده بیشتر از ۹۹۹ نخواهد بود و آن را چاپ می‌کند.

برای اینکه خروجی برنامه زیباتر باشد، آن را به این ترتیب چاپ می‌کنیم که عدد را نوشته و به تعداد ارقامی که نیاز است تا عدد ما به حداکثر ارقام مجاز رمز برسد در سمت راست آن صفر می‌گذاریم. به طور مثال اگر حداکثر ارقام مجاز ۵ رقم و عدد تولید شده ۱۰۹ باشد می‌نویسیم:

00109

برنامه را نوشته و در مورد آن با مدرس خود صحبت کنید.

حال فرض کنید که رمز تولید شده را در متغیر pass و حداکثر تعداد ارقام مجاز را در متغیر n ذخیره کرده‌اید. دستور زیر را برای چاپ خروجی استفاده کنید:

```
printf("%0*d\n", n, pass);
```

در مورد آن با مدرس کارگاه خود صحبت کنید.

امتیازی. فرض کنید n سیم روی پشت‌بام قرار دارند که با ۱ تا n شمارگذاری شده‌اند. این سیم‌ها تا زیرزمین کشیده شده‌اند ولی ما نمی‌دانیم سر سیم‌های زیرزمین کدام سیم پشت‌بام می‌باشد. برای این منظور ما می‌توانیم سیم‌های پشت‌بام را به هم گره بزنیم و با اهم متر در زیرزمین ببینیم که انتهای چه سیم‌هایی در پشت‌بام به یکدیگر متصل شده‌اند. به نظر شما می‌توان با این روش ۲ سیم را تفکیک کرد؟ ۳ سیم را چطور؟ ۴ سیم را چطور؟ آیا یک روند کلی می‌توانید پیشنهاد دهید؟

در صورتی که یکبار به پشت‌بام رفتن و برگشتن معادل یک گام باشد، روش‌های پیشنهادی خود را بر اساس تعداد گام با یکدیگر مقایسه کنید.

برای رسیدن به راحل نهایی می‌توانید سیم‌ها را به دو دسته تقسیم کنید و با وصل کردن تمام سیم‌های یک دسته به یکدیگر دسته‌ها را از یکدیگر متمایز کنید. حال می‌توانید هر دسته را به روش بازگشتی حل کنید.