

دانشگاه بین المللی امام خمینی



IMAM KHOMEINI
INTERNATIONAL UNIVERSITY

دانشگاه بین المللی امام خمینی قزوین

دانشکده فنی و مهندسی

مبانی کامپیوتر و برنامه سازی

(دومین ترم کرونا!)

فصل پنجم: ورودی و خروجی

نستوه طاهری جوان

nastoooh@aut.ac.ir



ورودی و خروجی در زبان C

- در زبان C روش های گوناگونی برای ورود و خروج اطلاعات وجود دارد.
- دستگاه ورودی پیش فرضمان در این درس، صفحه کلید است.
- دستگاه خروجی پیش فرضمان در این درس، مانیتور است.



ورودی و خروجی در زبان C

○ تابع printf()

- تعریف این تابع در فایل سرآیند `stdio.h` قرار دارد.
- یادآوری: معنی جمله بالا این است که برای استفاده از تابع `printf` در یک برنامه، باید حتما ابتدا دستور زیر را به ابتدای برنامه افزود:

`#include <stdio.h>`

- الگوی استفاده از این تابع به صورت زیر است:

`printf (“رشته کنترلی”);` (آرگومانها ,

- آرگومانها اطلاعاتی هستند که باید به خروجی منتقل شوند. (مثلا متغیرها)
 - آرگومانها می توانند وجود نداشته باشند.
- رشته کنترلی می تواند شامل سه بخش باشد:
 - اطلاعاتی که باید عیناً در خروجی چاپ شوند.
 - کاراکترهای تعیین کننده فرمت خروجی (شاخص های تبدیل)
 - کاراکترهای کنترلی

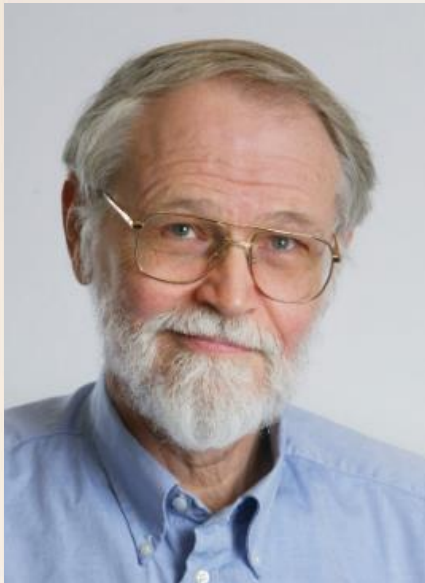


ورودی و خروجی در زبان C

○ تابع printf()

- مثال: اولین برنامه قابل اجرای ما:

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



- رسم چاپ "سلام دنیا" به عنوان اولین برنامه به هر زبان
➤ یک تفاهم نامه داخلی در آزمایشگاههای بل،
ابداع شده توسط آقای برایان کرنیگان!



ورودی و خروجی در زبان C

○ یادآوری چند نکته:

- نیاز به استفاده از یک کامپایلر برای ترجمه برنامه های زبان C به زبان ماشین
 - وجود کامپایلرهای مختلف و متفاوت برای زبان C
 - برنامه های زبان C را می توان توسط هر نرم افزار ویرایشگر متنی نوشت. مثلا Notepad
 - بعد از نوشتن برنامه و ذخیره آن در یک فایل، باید آن را کامپایل کرد.
 - فایل اجرایی حاصل از کامپایل را می توان اجرا کرد.
 - برای نوشتن، ویرایش، خطایابی، کامپایل و اجرای برنامه های نوشته شده، نرم افزارهایی با عنوان (IDE (Integrated Development Environment نیز وجود دارد. (که کار برنامه نویسی را بسیار راحتتر می کنند!
- مانند Microsoft Visual Studio (برای اجرا بر روی کلیه سکوهای ویندوزی)
 - یا Mono-Develop برای لینوکس
 - یا Xmac برای سیستم عامل مک
 - یا C++ Builder
 - یا Eclipse



ورودی و خروجی در زبان C

○ پیشنهاد ما برای IDE

• نرم افزار Code::Blocks

➤ اوپن سورس و مجانی

➤ داشتن امکانات خوب برای کمک به خوانایی برنامه ها

» رنگی کردن بخش هایی از کد

» کمک به تو رفتگی ها

» پلاگین های متنوع و و و

➤ [لینک دانلود](#) (برای ویندوز ۶۴ بیتی)

• یا نرم افزار Eclipse

➤ باز هم اوپن سورس و مجانی...

➤ داشتن امکانات خوب

➤ نیاز به نصب java runtime environment

➤ لینک دانلود

• یا هر IDE یا کامپایلر دیگری



ورودی و خروجی در زبان C

○ یادآوری نکته مهم در برنامه نویسی:

Indentation (تورفتگی، دندانه گذاری)

• کمک به خوانایی برنامه

```
...  
...  
{  
    ...  
    ...  
    {  
        ...  
        ...  
        ...  
    }  
    ...  
    ...  
}
```



ورودی و خروجی در زبان C

○ تابع printf()

• مثال:

```
printf ("My name is Nastooh");
```

My name is Nastooh

نتیجه خروجی:

• مثال:

```
printf ("My name is ");
```

```
printf ("Nastooh");
```

My name is Nastooh

نتیجه خروجی یکسان:



ورودی و خروجی در زبان C

○ تابع printf()

- کاراکترهای کنترلی: برای تعیین فرمت خروجی بکار می روند.
- همواره با بک-اسلش (\) می آیند.
- مثلا: \n در تابع printf() موجب انتقال مکان نما به خط بعد می شود.

➤ مثال:

```
printf ("My name is\n Nastooh");
```

نتیجه:

```
My name is
Nastooh
```

و یا حالت زیر:

```
printf ("My name is\n");
printf ("Nastooh");
```



ورودی و خروجی در زبان C

○ تابع printf()

- تعدادی از کاراکترهای کنترلی پر کاربرد:

کاراکتر کنترلی	نتیجه
\n	خط جدید
\t	tab
\"	چاپ "
\'	چاپ '
\\	چاپ \

- مثال:

```
printf ("My name is \"ALI\" ");
```

نتیجه:

My name is "ALI"



ورودی و خروجی در زبان C

○ تابع printf()

- یادآوری: الگوی تابع printf():

printf (“رشته کنترلی” , آرگومانها);

- آرگومانها، عبارات یا متغیرهایی هستند که باید مقدار آنها در خروجی چاپ شود.
- به ازای هر آرگومان باید یک شاخص تبدیل در جای مناسب در رشته ی کنترلی وجود داشته باشد.
- شاخص های تبدیل با علامت % می آیند.
- شاخص های تبدیل، در واقع تفسیر مناسب برای چاپ را از آرگومان مشخص می کنند.
- مثال:

```
int i = 15;
printf (“%d”, i);
```

نتیجه:

15



ورودی و خروجی در زبان C

○ تابع `printf()`

• چند شاخص تبدیل پر کاربرد:

فرمت	شاخص تبدیل
عدد دهدهی	%d
تک کاراکتر	%c
عدد اعشار ممیز شناور	%f
رشته	%s
عدد مبنای هشت	%o
عدد مبنای شانزده	%x
عدد صحیح علامتدار	%i
چاپ علامت درصد	%%



ورودی و خروجی در زبان C

○ تابع printf()

• مثال:

```
int i = 120;  
float f = 5.5;  
printf(“%d%f”, i, f);
```

نتیجه خروجی:

1205.500000

شاخص تبدیل %f به صورت پیش فرض تا ۶ رقم اعشار نشان می دهد

• مثال:

```
int i = 120;  
printf(“i=%d”, i);
```

نتیجه خروجی:

i=120



ورودی و خروجی در زبان C

○ تابع printf()

- نکته: پیشنهاد می شود جهت خوانایی برنامه، از بکار بردن شاخص های تبدیل ناهمگن اجتناب کنید. اما کماکان بهتر است خروجی آنها را به درستی حدس بزنید
- مثال:

```
int i = 65;
printf(“%d %c”, i, i);
```

نتیجه:

65 A

- مثال:

```
int i = 65;
printf(“%f”, i );
```

نتیجه:

خروجی غیر قابل پیش بینی



ورودی و خروجی در زبان C

○ تابع printf()

- نکته: تعداد شاخص های تبدیل و آرگومانها باید برابر باشند.
- اگر آرگومانها بیشتر باشند، فقط و فقط به تعداد و ترتیب شاخص های تبدیل داده چاپ می شود.
- اگر شاخص های تبدیل بیشتر باشند، به ازای شاخص های تبدیل اضافه، داده های غیر قابل پیش بینی نمایش داده می شود.

• مثال:

```
int i = 6, j = 7, k = 8;
printf(“%d %d\n”, i, j, k);
printf(“%d %d”, i);
```

نتیجه:

```
6 7
6 485892
```



ورودی و خروجی در زبان C

○ تابع printf()

• در آرگومانها می توان از عبارت محاسباتی استفاده کرد.

• مثال:

```
int i=10, j=4;  
printf(“%d %d %d”, i, i * j, j + i - 1);
```

نتیجه:

10 40 13



ورودی و خروجی در زبان C

○ تابع printf()

- نکته: اگر در آرگومانها عبارت محاسباتی وجود داشت، عملیات محاسبه (فارغ از چاپ) به ترتیب از عبارت سمت راست شروع می شود و سپس در صورت نیاز عبارت سمت چپ از نتیجه آن استفاده می کند!!!

- مثال:

```
int i = 0, j = 0;  
printf(“%d %d”, i = j + 2, j = i + 2);
```

نتیجه:

4 2



ورودی و خروجی در زبان C

○ تابع scanf()

- تعریف این تابع در فایل سرآیند `stdio.h` قرار دارد.
- یادآوری: معنی جمله بالا این است که برای استفاده از تابع `scanf` در یک برنامه، باید حتما ابتدا دستور زیر را به ابتدای برنامه افزود:

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

- الگوی استفاده از این تابع به صورت زیر است:

scanf (“رشته کنترلی” , آدرس آرگومانها ,

- آرگومانها در عمل متغیرهایی هستند که باید داده هایشان از ورودی خوانده شود. رشته کنترلی می تواند شامل سه بخش باشد:
 - کاراکترهای تعیین کننده فرمت (شاخص های کنترل)
 - کاراکترهای فضای خالی
 - کاراکترهای غیر فضای خالی



ورودی و خروجی در زبان C

○ تابع scanf()

• مثال:

```
#include<stdio.h>
void main( )
{
    int i;
    scanf ("%d", &i);
    printf ("%d", i + 1);
}
```

• نکته: هنگام اجرا، پس از ورود مقدار برای i باید حتماً اینتر زده شود. حتماً...

نتیجه برنامه: ابتدا یک عدد را از صفحه کلید دریافت کرده، پس از فشردن اینتر، آن را در متغیر i ذخیره کرده و در نهایت عدد بعدی آن را نمایش می دهد.



ورودی و خروجی در زبان C

○ تابع scanf()

• مثال:

```
int i, j, s;  
scanf ("%d%d", &i, &j);  
s = i * j;  
printf ("%d", s);
```

نتیجه برنامه: دو عدد دریافت شده و حاصلضرب آنها نمایش داده می شود.

برای ورودی: 5 3

خروجی: 15

- برای جدا کردن اعداد ورودی موقع اجرا، از فاصله یا اینتر می توان استفاده کرد.
- موقع خواندن اعداد دهدهی (با %d) هر کاراکتر دیگری به غیر از ارقام 0 تا 9 می تواند جداکننده دو عدد ورودی برای دو متغیر مجزا باشد.



ورودی و خروجی در زبان C

○ تابع scanf()

- تابع scanf() به تعداد شاخص های کنترلی از ورودی داده دریافت می کند.
 - اگر تعداد آرگومانها از شاخص های کنترلی بیشتر بود، متغیرهای اضافی مقدار دهی نمی شوند.
 - اگر تعداد شاخص های کنترلی از آرگومانها بیشتر بود، داده های اضافی از ورودی دریافت شده، اما دور ریخته می شوند.
- مثال:

```
int i, j, s;
```

```
scanf ("%d%d", &i, &j, &s);
```

نتیجه: فقط دو عدد دریافت شده و متغیر S مقداردهی نمی شود.

- مثال:

```
scanf ("%d%d%d", &i, &j);
```

- نتیجه: سه مقدار از ورودی خوانده می شود. اما عدد سوم از بین می رود.



ورودی و خروجی در زبان C

○ تابع scanf()

- اگر هنگام خواندن ورودی، به کاراکتری غیر مجاز برسیم، خطایی صادر نمی شود، اما کار خواندن خاتمه می یابد.
- مثال:

```
int i;  
scanf ("%d", &i);  
printf ("%d", i);
```

نتیجه:

ورودی:

125w43

خروجی:

125



ورودی و خروجی در زبان C

○ توصیه برنامه نویسی:

- جهت کاربرپسند شدن برنامه ها، اکیداً توصیه می شود قبل از دریافت هر داده ای از کاربر، توضیح مناسبی برای وی نمایش داده شود.

```
printf("Please enter two integers:");  
scanf ("%d%d", &i, &j);
```



ورودی و خروجی در زبان C

○ کمی تمرین

- برنامه های فلوجارت های صفحات ۱۷ و ۱۸ و ۱۹ از اسلایه های فصل سوم را بنویسید.



ورودی و خروجی در زبان C

○ فرمت بندی خروجی در تابع printf()

- تعریف طول میدان: فضایی که هر قلم اطلاعاتی در خروجی اشغال می کند.
- تعیین طول میدان به منظم شدن خروجی کمک می کند.
- نحوه تعیین طول میدان اعداد صحیح در printf():

%wd

که عدد W طول میدان را مشخص می کند.

- اگر طول میدان از عدد بزرگتر باشد، عدد از راست در میدان تراز می شود.
- اگر طول میدان از تعداد ارقام عدد کمتر بود، طول میدان نادیده گرفته می شود.
- مثال:

```
int i = 23, j = 1234;
```

```
printf("num1=%3d, num2=%3d", i, j);
```

```
num1= 23, num2=1234
```

خروجی



ورودی و خروجی در زبان C

○ فرمت بندی خروجی در تابع `printf()`

- نحوه تعیین طول میدان اعداد اعشاری در `printf()`:

`%w.df`

- که عدد `w` طول کل میدان و عدد `d` تعداد ارقام اعشار است.
- اگر طول میدان بخش صحیح از تعداد ارقام صحیح بزرگتر باشد، به تعداد نیاز فضای خالی سمت چپ عدد نمایش داده می شود.
- اگر طول میدان بخش صحیح از تعداد ارقام صحیح کمتر باشد، نادیده گرفته می شود.
- اگر طول میدان بخش اعشار، از تعداد ارقام اعشار بیشتر باشد، به تعداد نیاز `0` سمت راست عدد چاپ می شود.
- اگر طول میدان بخش اعشار از تعداد ارقام اعشار کمتر باشد، عدد گرد می شود.
- کاراکتر ممیز (`.`) نیز یک خانه محسوب می شود.



ورودی و خروجی در زبان C

○ فرمت بندی خروجی در تابع printf()

• مثال:

```
float f=123.567;  
printf("num is:%9.4f", f);
```

نتیجه خروجی:

num is: 123.5670

• مثال:

```
float f=123.567;  
printf("num is:%4.2f", f);
```

نتیجه خروجی:

num is:123.57



ورودی و خروجی در زبان C

○ فرمت بندی خروجی در تابع `printf()`

- برای تراز نمودن از چپ از یک علامت منفی به صورت زیر استفاده می کنیم:

`%-wd`

`%-w.df`

- مثال:

```
float i = 23;  
printf("num is:%-5d", i);
```

num is:23□□□

نتیجه:



ورودی و خروجی در زبان C

○ تعیین طول داده هنگام ورود اطلاعات با scanf()

- در تابع scanf() می توان بعد از علامت % تعداد کاراکترها یا ارقامی که وارد می شوند را تعیین کرد

- مثال:

```
int i, j, k;  
scanf(“%3d%2d%1d”, &i, &j, &k);
```

نتیجه:

با ورودی:

12345678

مقادیر متغیرها:

i=123 j= 45 k=6



ورودی و خروجی در زبان C

○ ورودی فرمت بندی شده با scanf()

- در تابع scanf() می توان بین شاخص های تبدیل، از کاراکترهایی استفاده کرد، در این صورت هنگام اجرا باید دقیقا آن کاراکترها (به عنوان جداکننده) بین ورودیها وارد شوند! (این مقادیر دور ریخته می شوند)

• مثال:

```
int i, j;
scanf("%db%d", &i, &j);
```

نتیجه:

با ورودی:

12b34

مقادیر متغیرها:

i=12 j= 34



ورودی و خروجی در زبان C

○ تمرین های تکمیلی printf():

- برای تعیین طول میدان اعداد صحیح، در حین اجرا (به صورت متغیر و نه ثابت) در تابع printf() چه تمهیدی اندیشیده شده است؟ با ذکر مثال کاملا تشریح کنید.
➤ راهنمایی:

%*d

- برای تعیین طول میدان اعداد اعشار، در حین اجرا (به صورت متغیر و نه ثابت) در تابع printf() چه تمهیدی اندیشیده شده است؟ با ذکر مثال کاملا تشریح کنید.
➤ راهنمایی:

%*.*f

- در مورد فرمت بندی و تعیین طول میدان در داده های رشته ای تحقیق کنید و با چند مثال آن را توضیح دهید.
➤ راهنمایی:

%m.ns



ورودی و خروجی در زبان C

○ تمرین های تکمیلی printf():

• در تابع printf() کاراکترهای کنترلی زیر چه اعمالی انجام می دهند؟

\a

\b

\r

• در تابع printf() شاخص تبدیل %d به چه منظور استفاده می شود؟

• در تابع printf() شاخص تبدیل %d به چه منظوری استفاده می شود؟

• در تابع printf() شاخص تبدیل %e و %E به چه منظور استفاده می شود؟



ورودی و خروجی در زبان C

○ تمرین های تکمیلی printf():

- در تابع printf() شاخص تبدیل %g و %G به چه منظور استفاده می شود؟
- در تابع printf() شاخص تبدیل %09d چه تفاوتی با شاخص %9d دارد؟
- تفاوت میان شاخص های تبدیل %x و %X و %#x و %#X را هنگام نمایش اعداد صحیح به کمک printf() را با ذکر مثال تشریح کنید.



ورودی و خروجی در زبان C

○ تمرین های تکمیلی scanf():

- در تابع scanf()، شاخص تبدیل %C به چه منظوری استفاده می شود؟
- تفاوت میان شاخص های تبدیل %d و %i و %o و %x را هنگام خواندن اعداد صحیح به کمک scanf() را با ذکر مثال تشریح کنید.



ورودی و خروجی در زبان C

○ سه تابع نسبتاً مشابه `getch()` و `getche()` و `getchar()`

- هر سه تابع برای دریافت یک کاراکتر از ورودی به کار می رود.
- نحوه استفاده از هر سه تابع مشابه است و به دو صورت بکار می رود

`getch();`

یا

`getch() = متغیر`

- الگوی دو تابع `getch()` و `getche()` در فایل `conio.h` قرار دارد و الگوی تابع `getchar()` در فایل `stdio.h` قرار دارد.
- تفاوت آنها:
- `getch()`: عکس العملی در صفحه نمایش ندارد و کاراکتر وارد شده را نمایش نمیدهد.
- `getche()`: کاراکتر وارد شده را نمایش می دهد.
- `getchar()`: پس از ورود کاراکتر حتما نیاز به فشردن اینتر دارد.



ورودی و خروجی در زبان C

○ سه تابع نسبتاً مشابه `getch()` و `getche()` و `getchar()`

- یکی از کاربردهای این توابع، علاوه بر مقدار دهی متغیرهای کاراکتری، ایجاد مکث در اجرای برنامه تا زمان فشردن کاراکتری از طرف کاربر است.
- به عنوان مثال، موکداً توصیه می شود همواره در انتهای برنامه، به نحو مطلوبی یکی از این توابع را فراخوانی کرد تا نتیجه خروجی (قبل از بسته شدن برنامه) توسط کاربر مشاهده شده و کاربر با فشردن یک کلید (مثلاً اینتر) دستور بسته شدن برنامه را بدهد.

```
#include<stdio.h>
void main()
{
    .....
    .....
    .....
    getchar();
}
```



ورودی و خروجی در زبان C

○ تمرین

- در مورد توابع `putchar()` و `putch()` تحقیق کرده و با زدن مثال با آنها آشنا شوید.
➤ راهنمایی: اولی در فایل `conio.h` و دومی در فایل `stdio.h` قرار دارد.
- در مورد تابع `clrscr` تحقیق کنید.
➤ راهنمایی: الگوی آن در `conio.h` قرار دارد.
➤ راهنمایی: فقط در بعضی از کامپایلرها و IDE ها مانند Turbo C پیاده سازی شده است.



پایان