

مبانی برنامه نویسی

مدرس: سعدون عزیزی

s.azizi@uok.ac.ir

مرکز آموزش های الکترونیکی

تابستان ۹۶

سرفصل مطالب

- آشنایی با کامپیوتر و الگوریتم
- مقدمه‌ای بر برنامه‌نویسی C
- محاسبات
- ورودی/خروجی
- حلقه‌ها
- دستورات شرطی
- توابع
- آرایه‌ها
- کاراکترها و رشته‌ها
- اشاره‌گرها
- ساختار
- فایل‌ها

مقدمه

- در C به منظور انجام عملیات ورودی و خروجی از توابعی استفاده می شود که از قبل نوشته شده اند و مشخصات آنها در کتابخانه `stdio.h` قرار دارد
- تابع `printf`: برای مشاهده خروجی برنامه روی صفحه نمایش
- تابع `scanf`: دریافت داده ها از صفحه کلید
- تابع `getchar`: دریافت یک کاراکتر از صفحه کلید
- تابع `putchar`: نمایش یک کاراکتر روی صفحه نمایش
- تابع `gets`: دریافت یک رشته از صفحه کلید
- تابع `puts`: نمایش یک رشته روی صفحه نمایش

تابع printf

□ الگوی تابع printf

```
printf("control string", arg1, arg2, ..., argn);
```

مثال:

```
printf("a=%d and b=%d", 12, -5*5);
```

خروجی:

a=12 and b=-25

چاپ مقادیر عددی صحیح

کاراکتر کنترلی	مفهوم معادل
d	نمایش یک عدد صحیح در مبنای ۱۰
i	نمایش یک عدد صحیح در مبنای ۱۰ (در چاپ i و d معادل هستند)
o	نمایش یک عدد صحیح بدون علامت در مبنای ۸
u	نمایش یک عدد صحیح بدون علامت در مبنای ۱۰
X یا x	نمایش یک عدد صحیح بدون علامت در مبنای ۱۶
h	قبل از هر یک از کاراکترهای بالا می آید و برای نمایش عدد صحیح short است
l (حرف l)	قبل از هر یک از کاراکترهای بالا می آید و برای نمایش عدد صحیح long است

چاپ مقادیر عددی اعشاری

مفهوم معادل	کاراکتر کنترلی
نمایش یک عدد اعشاری	f
نمایش یک عدد اعشاری به صورت نمایی (نماد علمی)	E یا e
نمایش یک عدد اعشاری به صورت ساده یا نمایی	G یا g
قبل از هر یک از کاراکترهای بالا می آید و برای نمایش عدد اعشاری double است	l (حرف l)
قبل از هر یک از کاراکترهای بالا می آید و برای نمایش عدد اعشاری long double است	L

چاپ مقادیر کاراکتری و رشته ای

مفهوم معادل	کاراکتر کنترلی
نمایش یک مقدار کاراکتری	C
نمایش یک مقدار رشته ای	S

فرمت خروجی

□ برای تعیین طول میدان از یک عدد صحیح مثبت بین علامت % و کاراکتر کنترلی استفاده می شود

```
printf(“***%d***\n”,12345);
```

```
printf(“***%8d***\n”,12345);
```

خروجی:

```
***12345***
```

```
*** 12345***
```

□ اگر طول میدان به اندازه کافی نباشد، مقدار آن تاثیری روی فرمت خروجی ندارد.

فرمت خروجی

□ در چاپ مقادیر عددی، علامت ها (+ و -)، ممیز (.) و نما (e یا E) هر کدام یک مکان به حساب می آیند

```
printf(“***%8d***\n”,-12345);
```

```
printf(“***%12f***\n”,123.45);
```

```
printf(“***%14e***\n”,123.45);
```

خروجی:

```
*** -12345***
```

```
*** 123.450000***
```

```
*** 1.234500e+02***
```

فرمت خروجی

```
printf(“***%5c***\n”,’m’);  
printf(“***%12s***\n”,”test line”);
```

خروجی:

```
***   m***  
*** test line***
```

فرمت خروجی

□ برای تعیین میزان دقت در چاپ، باید یک عدد صحیح مثبت را پس از یک نقطه اعشار بین علامت % و کاراکتر کنترلی مربوطه قرار داد

```
printf(“***%.4d***\n”,456);  
printf(“***%.2f***\n”, 123.456);  
printf(“***%.2e***\n”,123.456);  
printf(“***%.4s***\n”,”test line”);  
printf(“***%8.2f***\n”,123.456);
```

خروجی:

```
***0456***  
***123.46***  
***1.23e+02***  
***test***  
*** 123.46***
```

فرمت خروجی

مفهوم معادل	کاراکتر نمایشی
چاپ مقدار در منتهی علیه چپ میدان	- (منها)
چاپ علامت + قبل از عدد مثبت	+
نمایش 0 قبل از چاپ عدد در مبنای ۸ نمایش 0x یا 0X قبل از چاپ عدد در مبنای ۱۶	# (به همراه 0) (به همراه x یا X)
ستون های اضافی قبل از عدد در میدان با صفر پر می شوند	0 (صفر)

تابع scanf

□ الگوی تابع scanf

```
scanf("control string", &arg1, &arg2, ..., &argn);
```

مثال:

```
scanf("%d %f %c", &x, &y, &z);
```

□ برای خواندن رشته از ورودی، نیازی به علامت & قبل از متغیر نیست.

```
scanf("%s",str);
```

□ برای خواندن رشته از ورودی، می توان از [کاراکترهای مجاز] % و [کاراکترهای غیرمجاز] ^ % نیز استفاده کرد.

فرمت ورودی

□ تعیین طول میدان در خواندن

```
scanf(“%4d %3d”,&x, &y);  
printf(“x=%d and y=%d“,x,y);
```

1234567

x=1234 and y=567

□ ندیده گرفتن کاراکترها: برای دریافت تاریخ به صورت ۱۳۹۶/۴/۱ می توان از دستور زیر استفاده کرد

```
scanf(“%d/%d/%d”,&x, &y, &z);
```

توابع getchar، putchar و ...

□ تابع getchar: دریافت یک کاراکتر از صفحه کلید

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

□ تابع putchar: نمایش یک کاراکتر روی صفحه نمایش

```
putchar(c1);
```

□ تابع gets: دریافت یک رشته از صفحه کلید

```
gets(st1);
```

□ تابع puts: نمایش یک رشته روی صفحه نمایش

```
puts(st1);
```