دانشگاه بین المللی امام خمینی قزوین

دانشکدهٔ فنی و مهندسی

مبانی کامپیوتر و برنامه سازی

(دومین ترم کرونا!)

فصل هشتم: کاراکترها و رشته ها در C

نستوه طاهری جوان

nastooh@aut.ac.ir

مقدمه

- ✓ کاراکتر و رشته
  - تعریف: به دنباله ای از کاراکترها، رشته گویند.
  - مثال:
    - یک کاراکتر:

‘A’

- یک رشته:

“Ali”

  - کاراکترها در زبان C معمولا یک عدد صحیح (یک بایتی) در نظر گرفته می شوند، معادل کد ASCII آنها.

nastooh@aut.ac.ir

ورودی – خروجی کاراکترها

- یادآوری: در فصل های پیشین با دریافت و نمایش کاراکترها آشنا شدید:
  - چاپ کاراکتر با تابع printf() و شاخص تبدیل %c
  - چاپ کاراکتر با تابع putch()
  - چاپ کاراکتر با تابع putchar()
  - دریافت کاراکتر با تابع scanf() و شاخص تبدیل %c
  - دریافت کاراکتر با تابع getch()
  - دریافت کاراکتر با تابع getche()
  - دریافت کاراکتر با تابع getchar()

- یادآوری: تعریف و مقدار دهی اولیه به کاراکترها:

```
char  ch;
char  ch2 = ‘a’;
```


nastooh@aut.ac.ir

توابع کتابخانه ای ctype.h

در زبان C برای کارکردن با کاراکترها تعدادی تابع مفید وجود دارند که کار کردن با کاراکترها را آسان می کنند. الگوی این توابع در فایل ctype.h وجود دارند.

- تابع isdigit()
  - مشخص می کند یک کاراکتر جزو ارقام ’0‘ تا ’9‘ هست، یا خیر؟ اگر ورودی این تابع رقم باشد، مقدار یک و در غیر اینصورت مقدار صفر را بر میگرداند.

```
char  ch = ‘8’;
int  i;
i = isdigit (ch);
if (i = = 1)
    printf(“ch is digit!”);
```


nastooh@aut.ac.ir

توابع کتابخانه ای ctype.h

- تابع isalpha()
  - مشخص می کند یک کاراکتر جزو حروف الفبای بزرگ یا کوچک هست یا خیر؟ در صورتی که ورودی آن حروف کوچک ’a‘ تا ’z‘ و یا ’A‘ تا ’Z‘ باشد، خروجی آن عدد یک و در غیر اینصورت عدد صفر است.

```
char  ch = ‘A’;
int  i;
i = isalpha (ch);
if (i = = 1)
    printf(“ch is alphabet!”);
```

  - در بعضی از پیاده سازی ها از این تابع، اگر ورودی از حروف الفبای بزرگ باشد، خروجی آن ۱ و اگر از حروف کوچک باشد، خروجی آن ۲ است!

nastooh@aut.ac.ir

توابع کتابخانه ای ctype.h

- تابع islower()
  - مشخص می کند که ورودی آن یک حرف کوچک الفبا هست یا خیر؟ در صورتی که ورودی آن حروف کوچک ’a‘ تا ’z‘ باشد، خروجی آن عدد یک (در بعضی پیاده سازی ها دو) و در غیر اینصورت عدد صفر است.

- تابع isupper()
  - مشخص می کند که ورودی آن یک حرف بزرگ الفبا هست یا خیر؟ در صورتی که ورودی آن حروف کوچک ’A‘ تا ’Z‘ باشد، خروجی آن عدد یک و در غیر اینصورت عدد صفر است.

nastooh@aut.ac.ir

توابع کتابخانه ای ctype.h

- تابع tolower()
  - در این تابع اگر کاراکتر ورودیِ آن از حروف بزرگ الفبا باشد، حرف کوچک معادل آن را بر میگرداند. در غیر اینصورت همان کاراکتر را برمیگرداند.

```
char  ch1 = 'A',  ch2;
ch2 = tolower (ch1);
putch(ch2);
```

- تابع toupper()
  - در این تابع اگر کاراکتر ورودیِ آن از حروف کوچک الفبا باشد، حرف بزرگ معادل آن را بر میگرداند. در غیر اینصورت همان کاراکتر را برمیگرداند.

nastooh@aut.ac.ir

توابع کتابخانه ای ctype.h

- تمرین:
  - در مورد توابع زیر تحقیق کنید.

- isspace()
- isalnum()
- isxdigit()
- iscntrl()
- ispunct()
- isprint()
- isgraph()

nastooh@aut.ac.ir

مقدمه رشته ها

- در زبان C نوع جداگانه ای برای رشته ها وجود ندارد.
- برای این منظور از آرایه ای از کاراکترها استفاده می شود.
- آخرین خانه این آرایه باید با کاراکتر تهی (همان '\0') پر شود.
- مثال:

```
char  srt[10];
```

در این حالت رشته می تواند فقط ۹ کاراکتر باشد.

nastooh@aut.ac.ir

مقدمه رشته ها

- مقداردهی اولیه به رشته ها
  - روش اول:

```
char  str[10] = {‘A’, ‘L’, ‘I’, ‘\0’};
```

  - روش دوم:

```
char  srt[10] = “ALI”;
```

نتیجه هر دو روش در حافظه:

‘A’	‘L’	‘I’	‘\0’	?	?	?	?	?	?

- هنگام مقداردهی اولیه می توان طول رشته را مشخص نکرد، در این حالت طول آرایه یک واحد بیشتر از رشته خواهد شد. (برای ذخیره \0)

```
char  str[] = “IRAN”;
```


nastooh@aut.ac.ir

ورودی-خروجی رشته ها

- نمایش رشته در خروجی
  - با استفاده از تابع printf() و به کمک شاخص %s

```
char  str[10] = “Iran”;
printf(“%s”, str);
```

محتوای آرایه، کاراکتر به کاراکتر تا رسیدن به ’\0‘ نمایش داده می شود.

nastooh@aut.ac.ir

ورودی-خروجی رشته ها

- نمایش رشته در خروجی
  - با استفاده از تابع puts()
    - الگوی این تابع در فایل stdio.h قرار دارد.
    - نحوهٔ استفاده:

```
char  str[10] = “Iran”;
puts (str);

puts (“ali”);
```

- بعد از چاپ رشته با puts()، مکان نما به ابتدای خط بعد می رود!

nastooh@aut.ac.ir

ورودی-خروجی رشته ها

- خواندن رشته از ورودی
  - با استفاده از تابع scanf() و به کمک شاخص %s
  - **نکته مهم:** در این حالت نیازی به علامت & نیست.

```
char  str [20];
scanf(“%s”, str);
```

- **نکته مهم:** خواندن رشته تا یکی از کاراکترهای جداکننده ادامه می یابد. مثلا فاصله یا اینتر.
  - به عنوان مثال اگر برای ورودیِ مثال بالا عبارت hello world وارد شود، فقط عبارت hello در رشته ذخیره می شود!!

nastooh@aut.ac.ir

ورودی-خروجی رشته ها

- خواندن رشته از ورودی
  - با استفاده از تابع gets()
    - الگوی این تابع در stdio.h قرار دارد.
    - نحوهٔ استفاده:

```
char  str[20];
gets  (str);
```

- **نکته مهم:** در این حالت رشته می تواند شامل فاصله نیز باشد و خواندن رشته تا فشردن اینتر ادامه می یابد.

nastooh@aut.ac.ir

توابع کتابخانه ای string.h

برای کار کردن با رشته ها، توابع کتابخانه ای استانداردی وجود دارد که استفاده از این توابع، کار کردن با رشته ها را آسان می کند. الگوی این توابع در فایل string.h قرار دارد.

- تابع strlen()
  - یک مقدار صحیح را به عنوان طول رشته بر میگرداند.
  - تعداد کاراکترهای تا قبل از '\0' را می شمارد!
  - نحوهٔ استفاده با مثال:

```
int  i;
char  str[10] = “Iran”;
i = strlen(str);
```


nastooh@aut.ac.ir

توابع کتابخانه ای string.h

- تابع strcpy()
  - برای کپی کردن یک رشته در رشتهٔ دیگر استفاده می شود.
  - نحوهٔ استفاده:

```
strcpy (رشتهٔ مبدا , متغیر رشته ای);
```

  - مثال:

```
char  str1 [10];
char  str2 [10] = “Iran”;
strcpy (str1, str2);
strcpy (str1, “Ali”);
```

  - **نکته مهم**: به جز در مقدار دهی اولیه، استفاده از = برای انتساب رشته ها ممنوع است.

```
str1 = str2;
```
اشتباه است

  - نکته: اگر طول رشتهٔ مبدا بلندتر از رشتهٔ مقصد باشد، خطا رخ نمی دهد و کاراکترهای اضافی بلافاصله بعد از رشتهٔ مقصد قرار می گیرند و ممکن است متغییرهای دیگری ناخواسته تغییر کنند. مواظب باشید!

nastooh@aut.ac.ir

توابع کتابخانه ای string.h

- تابع strcat()
  - مقدار دو رشته را به هم می چسباند.
  - نحوهٔ استفاده:

strcat (رشتهٔ اول , رشتهٔ دوم);

  - مثال:

```
char  str1 [15] = “Iran”;
char  str2 [15] = “ALI”;
strcat (str1, str2);
```

- نکته: اگر طول رشتهٔ مقصد گنجایش اضافه شدن رشتهٔ دوم را نداشته باشد، خطا رخ نمی دهد و کاراکترهای اضافی بلافاصله بعد از رشتهٔ مقصد قرار می گیرند و ممکن است متغییرهای دیگری ناخواسته تغییر کنند. مواظب باشید!

nastooh@aut.ac.ir

توابع کتابخانه ای string.h

- تابع strcmp()
  - محتوای دو رشته را با هم مقایسه می کند. (نه طولشان را)
  - نحوهٔ استفاده:

strcmp (رشتهٔ اول , رشتهٔ دوم);

  - برای مقایسه، کدهای اسکی کاراکترها را به ترتیب مقایسه می کند.
  - خروجی آن یک عدد صحیح است:
    - اگر رشتهٔ اول بزرگتر بود، یک عدد مثبت برمیگرداند
    - اگر رشتهٔ اول کوچکتر بود، یک عدد منفی برمیگرداند
    - اگر دو رشته عیناً برابر بودند، صفر برمیگرداند
    - معمولا مقدار عددی برگشت داده شده، **اختلاف** کد اسکی **اولین اختلاف** رشته هاست.

```
char  str1 [15] = “computer”;
char  str2 [15] = “compiler”;
i = strcmp (str1, str2);
```

نتیجه در بسیاری از پیاده سازی ها: i=12

nastooh@aut.ac.ir

توابع کتابخانه ای string.h

- تابع strupr()
  - تمام حروف کوچک درون یک رشته را به حروف بزرگ تبدیل می کند.
  - نحوهٔ استفاده:

```
char  str[20] = “IrAn”;
strupr (str);
```

- تابع strlwr()
  - تمام حروف بزرگ درون یک رشته را به حروف کوچک تبدیل می کند.
  - نحوهٔ استفاده:

```
char  str[20] = “IrAn”;
strlwr (str);
```


nastooh@aut.ac.ir

توابع کتابخانه ای string.h

- تمرین:
  - برنامه ای بنویسید که اسامی ۱۰ دانشجو را از ورودی دریافت کرده و آنها را به ترتیب لیست کلاسی (بر اساس حروف الفبا) نمایش دهد.
    - آرایه ای از رشته ها تعریف کنید.
    - رشته ها را بخوانید.
    - آرایه اصلی را مرتب کنید. (مثلا به روش حبابی)
    - برای مقایسه رشته ها از strcmp استفاده کنید.

nastooh@aut.ac.ir

توابع کتابخانه ای string.h

- تمرین:
  - در بارهٔ توابع زیر تحقیق کنید.

- strrev()
- strcmpi()
- strncat()
- strncpy()
- strncmp()
- strncmpi()
- strstr()
- strchr()
- strrchr()
- memcpy()

nastooh@aut.ac.ir

توابع کتابخانه ای در stdlib.h

تعدادی تابع مفید برای کارکردن با رشته ها در فایل stdlib.h نیز وجود دارند.

- تابع strtod()
  - این تابع یک رشته را که حاوی کاراکترهای نشان دهندهٔ یک عدد اعشاری است را به یک مقدار عددی از نوع double تبدیل میکند.
    - برای کار کردن با این تابع باید با مفهوم اشاره گرها (در فصول بعد) آشنا باشید.

- تابع strtol()
  - این تابع یک رشته را که حاوی کاراکترهای نشان دهندهٔ یک عدد صحیح است را به یک مقدار عددی از نوع long int تبدیل میکند.
    - برای کار کردن با این تابع باید با مفهوم اشاره گرها (در فصول بعد) آشنا باشید.

nastooh@aut.ac.ir

توابع کتابخانه ای در stdio.h

علاوه بر توابع بررسی شده تا کنون، دو تابع مفید دیگر در فایل stdio.h برای کار با رشته ها وجود دارند.

- تابع sprintf()

- به کمک این تابع مفید، می توان یک داده عددی (یا فرمت بندی شده) را در یک رشته قرار داد.
- در این تابع می توان داده ها را فرمت بندی شده وارد رشته کرد.
- مثال:

```
char  s[30];
int  i = 453;
sprintf(s, “%d”,  i); // s= “453”
```


nastooh@aut.ac.ir

توابع کتابخانه ای در stdio.h

- تابع sprintf()
  - مثال:

```
char  s[30];
int  f = 45.3;
sprintf(s, “%f”,  f); // s= “45.30000”
```

  - مثال:

```
char  s[30];
int  i = 93, j = 3;
sprintf(s, “ i is %d and j is %d”,  i, j); // s= “i is 93 and j is 3”
```


nastooh@aut.ac.ir

توابع کتابخانه ای در stdio.h

- تابع sscanf()
  - به کمک این تابع میتوان از یک رشتهٔ کاراکتری، داده هایی را خواند!
  - مثال:

```
char  s[30] = “321  78  222”;
int  i,  j,  k;
sscanf(s, “%d%d%d”, &i,  &j,  &k); // i=321  j=78  k=222
```


nastooh@aut.ac.ir

منابع

[1] P. Deitel , H. Deitel, **C How to Program**, 8th ed., 2016. (Download Link)

[2] B. W. Kernighan, D. M. Ritchie, **The C Programming Language**, 2nd ed., 1988. (Download Link)

برای دانلود کتاب ها، اسلایدها و نمونه پروژه های درسی به سایت www.nastooh.com بخش دانشجویان مراجعه کنید.

پایان

nastooh@aut.ac.ir