

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

کاراکترها

```
char c1,c2,c3;  
printf("Enter two characters is separate lines:\n");  
scanf("%c%c%c",&c1,&c2,&c3);  
printf("c1=%c and c3=%c",c1,c3);  
printf("\n\nEnter three characters is separate lines:\n");  
scanf(" %c %c %c",&c1,&c2,&c3);  
printf("c1=%c and c3=%c",c1,c3);
```

نتیجه اجرا:

محیط ++VC De

کاراکترها

□ تابع های `getche` و `getch`، `getchar`

* توضیحات در محیط `DeVC++`

توابع مخصوص کاراکترها

- ❑ تابع **isdigit**: این تابع بررسی می کند که آیا آرگومان یکی از ارقام 0-9 می باشد؟
- ❑ تابع **isalpha**: این تابع بررسی می کند که آیا آرگومان یکی از حروف A-Z یا a-z می باشد؟
- ❑ تابع **isalnum**: این تابع بررسی می کند که آیا آرگومان یکی از حروف A-Z، a-z یا یکی از ارقام 0-9 می باشد؟ (کاراکتر حرفی-رقمی)
- ❑ تابع **isxdigit**: این تابع بررسی می کند که آیا آرگومان یکی از ارقام معتبر در مبنای ۱۶ می باشد؟
- ❑ تابع **isspace**: این تابع بررسی می کند که آیا آرگومان یکی از کاراکترهای جداکننده می باشد؟ (کاراکترهای جداکننده: '\v', '\r', '\f', '\t', '\n', '')

توابع مخصوص کاراکترها (ادامه)

- تابع **isupper** : این تابع بررسی می کند که آیا آرگومان یکی از حروف A-Z می باشد؟
- تابع **tolower** : این تابع بررسی می کند که آیا آرگومان یکی از حروف A-Z می باشد؟ چنانچه جواب درست بود، حرف کوچک معادل آن را باز می گرداند؛ در غیر این صورت همان کاراکتر بازگردانده می شود.
- تابع **toupper** : این تابع بررسی می کند که آیا آرگومان یکی از حروف a-z می باشد؟ چنانچه جواب درست بود، حرف بزرگ معادل آن را باز می گرداند؛ در غیر این صورت همان کاراکتر بازگردانده می شود.

رشته ها

□ برای ذخیره و بازیابی و ویرایش متن ها از رشته ها استفاده می شود

□ در بسیاری از زبان های برنامه نویسی برای رشته ها نوع مشخصی (معمولا string) در نظر گرفته می شود. اما در زبان C با رشته ها به صورت آرایه ای یک بعدی از کاراکترها برخورد می شود

□ مثال:

```
char str1[12]="C language";
```

str1

C	l	a	n	g	u	a	g	e	\0	\0
---	---	---	---	---	---	---	---	---	----	----

توابع مخصوص رشته ها

- تابع **strcpy(s1,s2)**: این تابع رشته s2 را در آرایه کاراکتری s1 کپی می کند و کاراکتر تهی ('0\') را پس از آخرین کاراکتر کپی شده قرار می دهد.
- تابع **strncpy(s1,s2,n)**: این تابع حداکثر n کاراکتر از رشته s2 را در آرایه کاراکتری s1 کپی می نماید. اگر n کوچکتر یا مساوی طول رشته s2 باشد همان n کاراکتر کپی می شود. در این حالت برنامه نویس باید به انتهای رشته کپی شده در s1 کاراکتر تهی را قرار دهد. اگر n بیش از طول رشته s2 باشد، بایت های باقی مانده با کاراکتر تهی پر می شوند.
- تابع **strcat(s1,s2)**: این تابع رشته s2 را به آرایه کاراکتری s1 متصل می نماید.
- تابع **strncat(s1,s2,n)**: این تابع حداکثر n کاراکتر از رشته s2 را به آرایه کاراکتری s1 متصل می نماید.

توابع مخصوص رشته ها

- تابع **strcmp(s1,s2)**: این تابع رشته s1 را با s2 مقایسه می کند. چنانچه دو رشته یکسان باشند مقدار صفر، اگر رشته s1 از نظر ترتیب الفبایی قبل از s2 قرار داشته باشد مقدار منفی و اگر s1 از نظر ترتیب الفبایی بعد از s2 قرار داشته باشد مقدار مثبت را برمی گرداند.
- تابع **strncmp(s1,s2,n)**: این تابع تا n امین کاراکتر از s1 را با s2 مقایسه می کند
- تابع **strcmpi(s1,s2)**: این تابع همانند تابع strcmp(s1,s2) عمل می کند با این تفاوت که نسبت به کوچک و بزرگ بودن حروف حساس نیست
- تابع **strncmpi**: این تابع همانند تابع strncmp(s1,s2,n) عمل می کند با این تفاوت که نسبت به کوچک و بزرگ بودن حروف حساس نیست
- تابع **strlen(s)**: این تابع طول رشته داده شده را بر می گرداند

ارسال رشته به تابع

□ برای ارسال رشته به تابع دقیقاً مانند سایر آرایه ها عمل می کنیم.

مثال در محیط ++VC De

آرایه ای از رشته ها

- برای ذخیره مجموعه ای از رشته ها از یک آرایه دو بعدی از نوع کاراکتر استفاده می کنیم
- در چنین آرایه ای، اندازه بعد اول تعداد رشته ها را مشخص می کند، در حالی که اندازه بعد دوم گنجایش (حداکثر طول) هر رشته را نشان می دهد

مثال در محیط ++VC++