

مبانی برنامه‌نویسی
(۱۱-۱۳-۱۳۹)
جلسه‌ی بیست و نهم

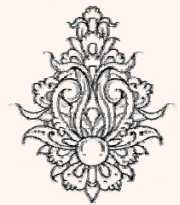
مرتب‌سازی
ارائه



دانشگاه شهید بهشتی
پاییز ۱۳۹۲
دانشکده‌ی مهندسی برق و کامپیوتر
احمد محمودی ازناوه

فهرست مطالب

- روش‌های مرتب‌سازی
- آشنایی با رشته



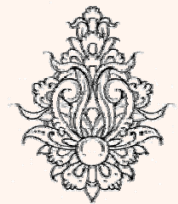
مرتب‌سازی انتخابی (صعودی)

	8
	5
	2
	6
	9
	3
	1
	4
	0
	7

wikipedia

- در این روش ابتدا اولین عنصر، با تمامی عناصرها مقایسه و کوچک‌ترین عنصر در نخستین خانه‌ی آرایه قرار می‌گیرد.
- سپس این روند، برای بقیه‌ی آرایه تکرار می‌شود.

```
void selectionsort(int k[],int l){
    int i,j,temp;
    for (i=0;i<l-1;i++)
        for (j=i+1;j<l;j++)
            if (k[i]>k[j]){
                temp=k[j];
                k[j]=k[i];
                k[i]=temp;
            }
}
```

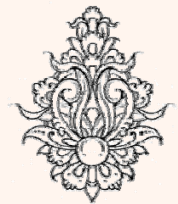


مرتب‌سازی انتخابی (ادامه...)

```
int main(int)
{
    const int n=10;
    int i,a[n];
    cout<<"enter ten integer number:\n";
    for(i=0;i<n;i++)
    {
        cout<<i+1<<"th:";
        cin>>a[i];
        cout<<"\n";
    }
    selectionSort(a,n);
    for(i=0;i<n;i++)
        cout<<a[i]<<"\t";
}
```

```
void selectionSort(int k[],int len)
{
    int Min,MinInd;
    for(int i=0;i<len-1;i++)
    {
        Min=k[i];
        MinInd=i;
        for(int j=i+1;j<len;j++)
            if(Min>k[j])
            {
                Min=k[j];
                MinInd=j;
            }
        int tmp=k[i];
        k[i]=Min;
        k[MinInd]=tmp;
    }
}
```

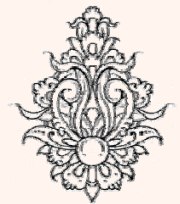
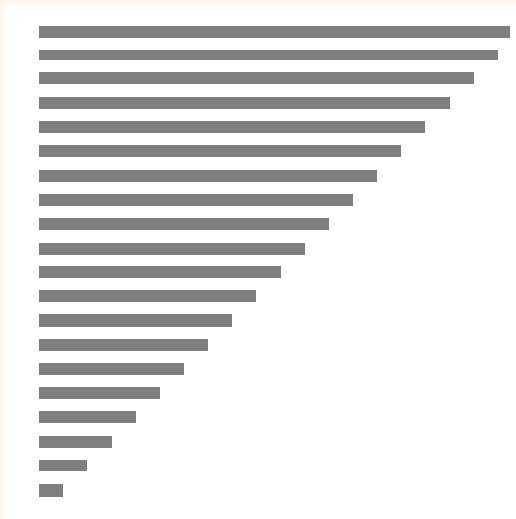
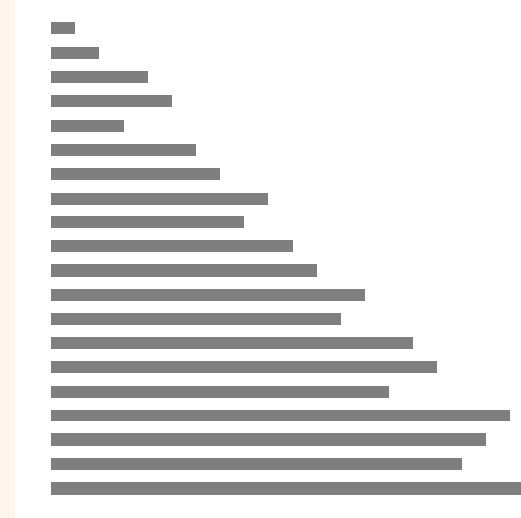
1th:8
2th:6
3th:4
4th:2
5th:1
6th:3
7th:9
8th:7
9th:5
10th:0



0 1 2 3 4 5 6 7 8 9

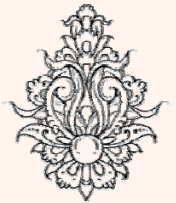
<http://www.sorting-algorithms.com/>

مرتب‌سازی انتخابی (ادامه...)



مرتب سازی مجابی

- «مرتب سازی مجابی» یکی از ساده ترین الگوریتم های مرتب سازی است.
- در این روش، آرایه چندین مرتبه پویش می شود و عناصر متوالی با هم مقایسه می شوند.
- در هر مرتبه کوچک ترین عنصر موجود به سمت بالا هدایت می شود، سپس از محدوده مرتب سازی برای مرتبه ی بعدی یکی گاسته می شود.
- در پایان تمامی آرایه مرتب شده است.



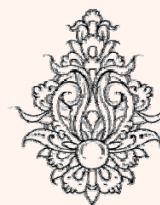
شیوه مرتب‌سازی جبابی

- (1) آخرین عنصر آرایه با عنصر قبل از خود مقایسه می‌شود.
- (2) اگر کوچک‌تر بود، جای این دو با هم عوض می‌شود.
- (3) سپس عنصر ماقبل آخر با عنصر قبل از خود مقایسه می‌شود.
- (4) اگر عنصر دوم بزرگ‌تر بود، جای این دو با هم عوض می‌شود.

و به همین ترتیب مقایسه و جابه‌جایی زوج‌های همسایه ادامه می‌یابد تا وقتی به ابتدای آرایه رسیدیم، کوچک‌ترین عضو آرایه در فانه‌ی نخست قرار خواهد گرفت.

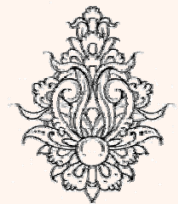
در ادامه **محدوده‌ی جستجو یکی کاسته می‌شود.**

به همین ترتیب زوج‌های کناری یکی یکی مقایسه می‌شوند تا عدد کوچک‌تر بعدی به جایگاه خود منتقل شود. این پویش ادامه می‌یابد تا این که وقتی محدوده جستجو به عنصر آخر محدود شد، آرایه مرتب شده است.



مثال

- برنامه‌ای بنویسید که یک آرایه ۱۰ عنصری را از ورودی دریافت کند، آن را به یک تابع مرتب‌سازی ارسال نماید و بعد از مرتب‌سازی آن را در تابع اصلی چاپ نماید.



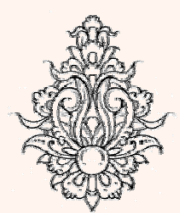
یاسر

```
#include <iostream>
using namespace std;
void bubblesort(int a[],int);

int main(int){
    const int n=10;
    int i,a[n];
    cout<<"enter ten integer number";
    for(i=0;i<n;i++){
        cout<<i+1<<"th:";
        cin>> a[i];
        cout<<"\n";
    }
    bubblesort(a,n);
    for(i=0;i<n;i++){
        cout<<a[i]<<"\n";
    }
}
```

```
void bubblesort(int k[],int l){
    int i,j,temp;
    for(i=0;i<l-1;i++){
        for(j=l-1;j>i;j--){
            if(k[j]<k[j-1]){
                temp=k[j];
                k[j]=k[j-1];
                k[j-1]=temp;
            }
        }
    }
}
```

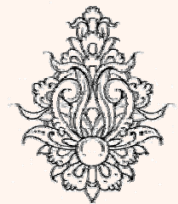
```
enter ten integer number:
1th:1
2th:4
3th:7
4th:4
5th:55
6th:8
7th:4
8th:1
9th:9
10th:33
```



مرتب‌سازی حبابی (ادامه...)

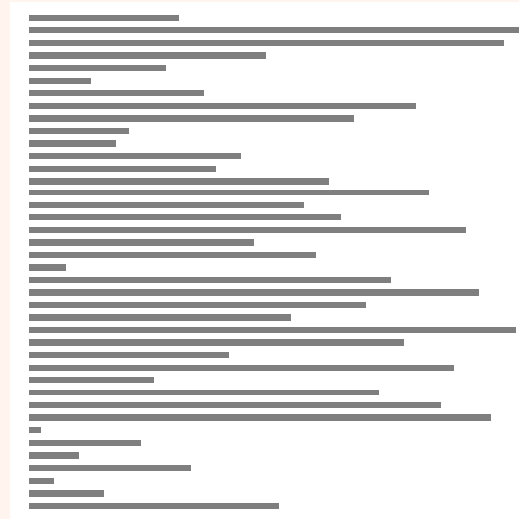
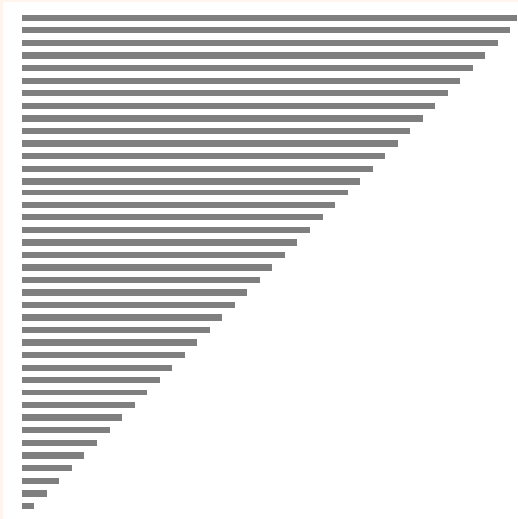
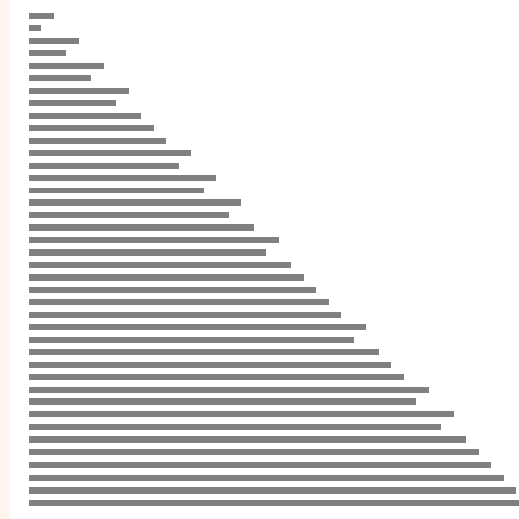
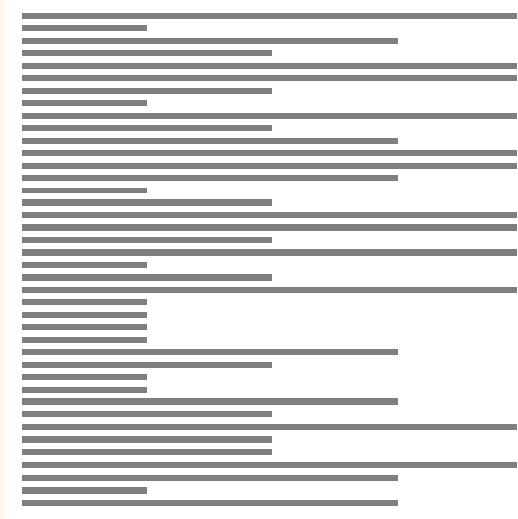
```
void bubblesort(int k[],int l){
    int i,j,temp;
    bool isSorted=false;
    for(i=0;i<l-1 && !isSorted;i++){
        isSorted=true;
        for(j=l-1;j>i;j--){
            if(k[j]<k[j-1]){
                isSorted=false;
                temp=k[j];
                k[j]=k[j-1];
                k[j-1]=temp;
            }
        }
    }
}
```

بدین ترتیب برای ارزیابی مرتب شده این حرکات
پایان ادامه نمی‌دهد.

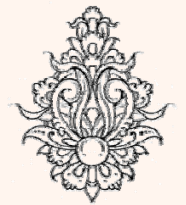


<http://www.sorting-algorithms.com/>

مرتب‌سازی حبابی (ادامه...)

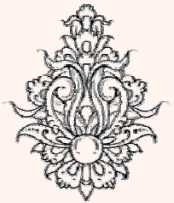


مبانی برنامه‌نویسی



رشته‌ها

- در زبان C++ یک «**رشته‌ی کاراکتری**» یک نوع داده نیست. در حقیقت رشته آرایه‌ای از کاراکترهاست که این آرایه دارای ویژگی مهم زیر است:
- در انتهای آرایه کاراکتر **NUL** که دارای ارزش صفر است وجود دارد. برای نمایش NUL از '\0' استفاده می‌شود. پس تعداد کل کاراکترها در آرایه همیشه یکی بیشتر از طول رشته است.
- رشته‌ی کاراکتری را می‌توان با لیترال رشته‌ای به طور مستقیم مقدارگذاری کرد مثل:
`char str[] = "string";`
توجه کنید که این آرایه هفت عنصر دارد:
'\0' و 'g' و 'n' و 'i' و 'r' و 't' و 's'



مقداردهی اولیه

- بدون تعیین حد

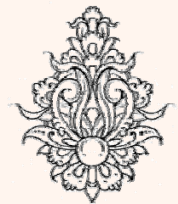
- Char s1[]="STRING";
- Char s2[]={ 'S','T','R','I','N','G','\0' };

- با تعیین حد

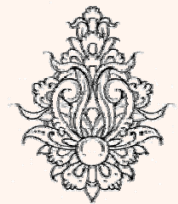
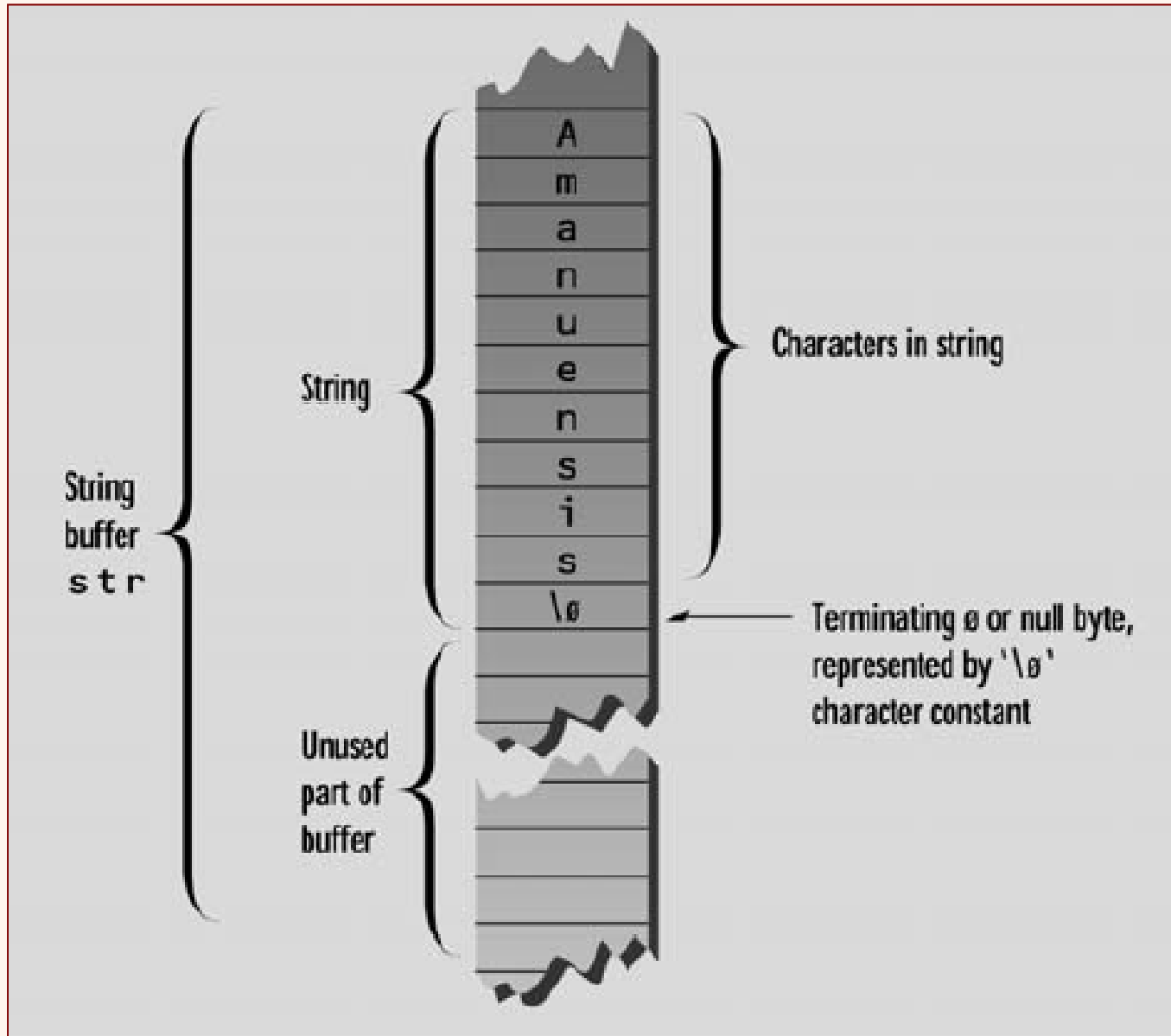
- Char s3[12]="STRING";
- Char s4[7]={ 'S','T','R','I','N','G','\0' };

- توابع ورودی و خروجی رشته‌ها:

<code>puts(s);</code>	<code>gets(s);</code>	- در زبان c
<code>cout<<s;</code>	<code>cin>>s;</code>	- در زبان c++



ذخیره‌ی رشته در حافظه



متغیر رشته‌ای

```
int main()  
{  
char str[] = "Look! This is a new sentences."  
cout << str << endl;  
return 0;  
}
```

Look! This is a new sentences.



مثال

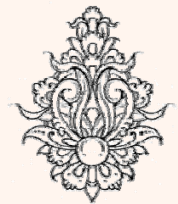
```
#include<iostream>
using namespace std;
int main()
{
    char s[5]="ABCD";
    for(int i=0; i< sizeof(s); i++)
        cout<<"s["<< i << "] = '" << s[i] << "'\n";
}
```

```
s[0] = 'A'
s[1] = 'B'
s[2] = 'C'
s[3] = 'D'
s[4] = ''
```

gcc

```
s[0] = 'A'
s[1] = 'B'
s[2] = 'C'
s[3] = 'D'
s[4] = ''
```

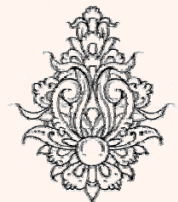
Microsoft Visual studio!



مثال

```
int main()
{
    char p[] = "Hello ";
    char q[] = "World!";
    cout << p << q << endl;
    cout << p << " is stored at " <<int(p)<< endl;
    cout << q << " is stored at " <<int(q)<< endl;
    cout<<int ("Another string literal")<<endl;
    return 0;
}
```

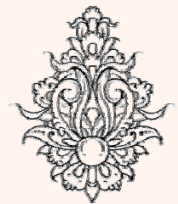
```
Hello World!
Hello  is stored at 1244960
World! is stored at 1244944
4290560
```



مثال

```
int main()
{
const int MAX = 80; //max characters in string
char str[MAX];
cout << "Enter a string: ";
cin >> str;
cout << "You entered: " << str << endl;
return 0;
}
```

```
Enter a string: this is a test
You entered: this
```

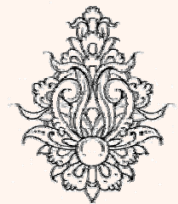


خواندن رشته

- اگر بخواهیم به وسیلهی `cin` یک جمله را از ورودی بخوانیم خواهیم داشت:

```
Enter a string: Law is a bottomless pit.  
You entered: Law
```

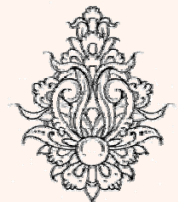
Space به عنوان پایان رشته تلقی می‌شود



مثال

```
int main()
{
const int MAX = 20;
char str[MAX];
cout << "\nEnter a string: ";
cin >>str;
cout << "You entered: " << str << endl;
return 0;
}
```

```
Enter a string: abcdefghijklmnopqrstuvwxyz
You entered: abcdefghijklmnopqrstuvwxyz
Segmentation fault
```



تابع get

– خواندن رشته با تابع get از «کلاس» cin

```
cin.get();
```

```
cin.get(c);
```

```
cin.get(s,15);      cin.getline(s,15);
```

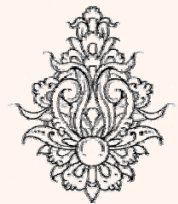
```
cin.get(s,15,'.');    cin.getline(s,15,'.');
```

– تفاوت با «cin.get»

– رشته‌ی گرفته شده به وسیله‌ی cin.get می‌تواند شامل Space و یا tab باشد.

– در cin این دو کاراکتر نیز آخر رشته را مشخص می‌کنند.

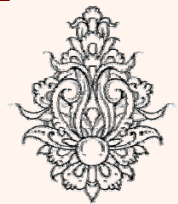
– برای چاپ خروجی از puts و یا cout استفاده می‌شود.



مثال

```
int main()
{
const int MAX = 80; //max characters in string
char str[MAX]; //string variable str
cout << "\nEnter a string: ";
cin.get(str, MAX); //put string in str
cout << "You entered: " << str << endl;
return 0;
}
```

```
Enter a string: this is a test
You entered: this is a test
```



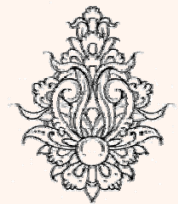
تمرین

برنامه‌ای بنویسید که تعدادی کاراکتر از ورودی گرفته و تعداد e را بشمارد.

Windows/Visual Studio

```
#include <iostream>
using namespace std;
int main(){
    char ch;
    int count = 0;
    while ((ch=cin.get()) != EOF)
        if (ch == 'e') ++count;
    cout << count << " e's were counted.\n";
}
```

```
This is a test. you are testing the function related to string
^Z
6 e's were counted.
```



در محیط windows . Ctrl+z و در محیط Linux . Ctrl+D نشانه‌ی EOF هستند.



تمرین

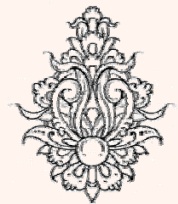
برنامه‌ای بنویسید که تعدادی کاراکتر از ورودی گرفته و تعداد e را بشمارد.

```
#include <iostream>
#include <cstdio>
using namespace std;
int main(){
    char ch;
    int count = 0;
    while ((ch=cin.get())!=EOF)
        if (ch == 'e') ++count;
    cout << count << " e's were counted.\n";
}
```

Linux/gcc

```
This is a test.1 e's were counted.
```

در محیط windows . Ctrl+z و در محیط Linux . Ctrl+D نشانه‌ی EOF هستند.



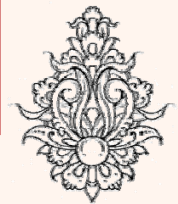
مثال

برنامه‌ای بنویسید که تعدادی کاراکتر از ورودی گرفته و تعداد e را بشمارد.

Linux/gcc

```
#include <iostream>
using namespace std;
int main(){
    char ch;
    int count = 0;
    while (cin.get(ch))
        if (ch == 'e') ++count;
    cout << count << " e's were counted.\n";
}
```

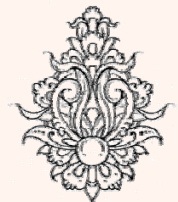
```
This is a test.
1 e's were counted.
```



مثال

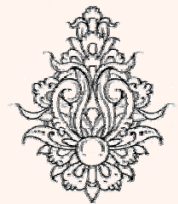
```
#include <iostream>
using namespace std;
const int MAX = 2000; //max characters in string
char str[MAX]; //string variable str
int main()
{
cout << "\nEnter a string:\n";
cin.get(str, MAX, '$');
cout << "You entered:\n" << str << endl;
return 0;
}
```

```
Enter a string:
this is a test.
you can enter multiple line here.
end of the text will be shown by a sign.
$
You entered:
this is a test.
you can enter multiple line here.
end of the text will be shown by a sign.
```



مثال

- برنامه‌ای بنویسید که رشته‌ای را به عنوان ورودی گرفته سپس تعداد کلمات رشته را چاپ نماید. (فرض بر این است که رشته با نقطه به پایان می‌رسد و بین «.» و کلمه‌ی پایانی فاصله وجود ندارد.)



```

#include <iostream>
using namespace std;
int main(){
    char str[100];
    int i=0, NOW=0;
    cin.get(str,100, '.');
    while (str[i] != '\0') {
        if (str[i] == ' ' || str[i] == '\t' || str[i] == '\n')
            ++NOW;
        i++;
    }
    cout<<"The number of the words:"<<++NOW<<endl;
}

```



```

it is a test
for programming class.
The number of the words:7

```



مثال

- هنگام کار با رشته فرآیند بر روی آرایه‌ای از کاراکترها صورت می‌پذیرد.
- برنامه‌ای بنویسید که رشته‌ای را در رشته‌ی دیگر کپی کند:

```
int main()
{
    int j;
    char str1[] = "This is a test! ";

    const int MAX = 80;
    char str2[MAX];
    for(j=0; str1[j]!='\0'; j++)
        str2[j] = str1[j];
    str2[j] = '\0';
    cout << str2 << endl;
    return 0;
}
```

This is a test!

