

Масиви и символни низове

Калин Георгиев

24 октомври 2024 г.

Първоначални сведения за масиви и низове

Представяне на низове

- ASCII таблица

...	A ⁶⁵	B ⁶⁶	C ⁶⁷	...
-----	-----------------	-----------------	-----------------	-----

- Представяне в паметта

...	H	E	L	L	O	_	...
...	72	69	76	76	79	0	...

- Масиви

Представяне на низове

- ASCII таблица

...	A ⁶⁵	B ⁶⁶	C ⁶⁷	...
-----	-----------------	-----------------	-----------------	-----

- Представяне в паметта

...	H	E	L	L	O	_	...
...	72	69	76	76	79	0	...

- Масиви

Представяне на низове

- ASCII таблица

...	A ⁶⁵	B ⁶⁶	C ⁶⁷	...
-----	-----------------	-----------------	-----------------	-----

- Представяне в паметта

...	H	E	L	L	O	_	...
...	72	69	76	76	79	0	...

- Масиви

Масиви

- Дефиниране чрез тип и размер:

```
int arr[100];
```

- Достъп до всеки отделен елемент:

```
int b = arr[18];  
cin >> arr[18];  
b = arr[1] + arr[2];
```

- Обхождане с for цикъл

```
for (int count = 0; count < 100; count++)  
{  
    cout << arr[count];  
}
```

Масиви

- Дефиниране чрез тип и размер:

```
int arr[100];
```

- Достъп до всеки отделен елемент:

```
int b = arr[18];  
cin >> arr[18];  
b = arr[1] + arr[2];
```

- Обхождане с for цикъл

```
for (int count = 0; count < 100; count++)  
{  
    cout << arr[count];  
}
```

Масиви

- Дефиниране чрез тип и размер:

```
int arr[100];
```

- Достъп до всеки отделен елемент:

```
int b = arr[18];  
cin >> arr[18];  
b = arr[1] + arr[2];
```

- Обхождане с for цикъл

```
for (int count = 0; count < 100; count++)  
{  
    cout << arr[count];  
}
```


Прост пример с низове

```
int main ()
{
    char str[100] = "Hello_world!";
    cout << str << endl;

    str[0] = 'Y';
    cout << str << endl;

    cout << "Please_input_a_string:";
    cin >> str;

    for (int counter = 0; counter < 100; counter++)
    {
        if (str[counter] == 'a')
        {
            str[counter] = 'b';
        }
    }

    cout << str << endl;

    return 0;
}
```

Какво не можем да правим с масиви и низове

- Няма проверка за коректност!

```
char a[6] = "HELLO";  
a[10] = '!';
```

...	0	1	2	3	4	5	...	10	...
...	H	E	L	L	O	_	...	!	...

- Присвояване ($a=b$)
- Сравнение ($a==b$, $a < b, \dots$)

Какво не можем да правим с масиви и низове

- Няма проверка за коректност!

```
char a[6] = "HELLO";  
a[10] = '!';
```

...	0	1	2	3	4	5	...	10	...
...	H	E	L	L	O	_	...	!	...

- Присвояване ($a=b$)
- Сравнение ($a==b$, $a < b, \dots$)

Какво не можем да правим с масиви и низове

- Няма проверка за коректност!

```
char a[6] = "HELLO";
a[10] = '!';
```

...	0	1	2	3	4	5	...	10	...
...	H	E	L	L	O	_	...	!	...

- Присвояване ($a=b$)
- Сравнение ($a==b$, $a < b, \dots$)

Вградени функции за работа с низове

Дължина на низ

```
cout << strlen(a);
```

Присвояване на низове

```
strcpy (d,a);
strcpy (c,a); //!!!
```

Сравнение на низове

```
if (strcmp (a,b) < 0)
{cout << "a<b";}
else
{cout << "b<=a";};
```

Конкатенация на низове

```
strcpy (d,a); //d -> "HELLO"
```

...	0	1	2	3	4	5	6	7	8	9	10	...
...	H	E	L	L	O	_	?	?	?	?	?	...

```
strcat (d,b); //d -> "HELLOWORLD"
```

...	0	1	2	3	4	5	6	7	8	9	10	...
...	H	E	L	L	O	W	O	R	L	D	_	...

```
#include <cstring>
```

```
...
char a[6] = "HELLO";
char b[6] = "WORLD";
char c[4] = "BYE";
char d[11] = "???";
```

...	0	1	2	3	4	5	...
...	H	E	L	L	O	_	...

...	0	1	2	3	4	5	...
...	W	O	R	L	D	_	...

...	0	1	2	3	...
...	B	Y	E	_	...