Низове

гл. ас. д-р. Нора Ангелова

Низове
(като масиви от символи)

```
char str1[10] = "123456789";
char str2[4] = "abc";
strcpy(str1, str2);
cout << str1;

Изход:
abc
```

Низове
(като масиви от символи)

```
char str1[10] = "0123456789";
char str2[4] = "abc";
strcpy(str1, str2);
cout << str1;
```

Изход:

ERROR – const char[11]

Низове
(като масиви от символи)

Символен низ е съставен единствено от малки латински букви.
Да се напише програма, която намира и извежда на екрана броя на срещанията на всяка от буквите на низа.

```
#include <iostream.h>
#include <string.h>
const int MAX_SIZE = 100;
int main()
{ char s[MAX_SIZE];
  unsigned int histogram[26] = {0},
               i = 0;
  cout << "s= ";
  cin >> s;
  for(i = 0; i < strlen(s); i++)
    histogram[s[i]-'a']++;
  for(i = 0; i < 26; i++)
    if(histogram[i] > 0)
      cout << (char)('a'+i) << ": "
           << histogram[i] << endl;
  return 0;
}
```

Низове
(като масиви от символи)

Името на променлива от тип низ е **константен** указател.

```
char str[7] = "123456";

while(*str) {
  cout << *str << " ";
  str++;                          // Error
}
```

Низове
(като масиви от символи)

Името на променлива от тип низ е **константен** указател.

```
// Указателят str е константен и стойността му не може да се променяна
// ВАЖИ И ЗА ВСЕКИ ДРУГ МАСИВ
char str[7] = "abcdef";
char *p = str;

while(*p) {
  cout << *p << " ";
  p++;                    // a b c d e f
}
```

Низове
(като масиви от символи)

Указател към символ от низа

```
char str[7] = "abcdef";
char *p = str;


cout << p+3; // def
```

Низове

(като указатели от тип char)

Низове
(като указател от тип char)

```
const char *str = "123456";

while(*str) {
  cout << *str << " ";
  str++;                          // 1 2 3 4 5 6
}
```

Низове
(като указател от тип char)

** Не е възможно въвеждане на стойност от тип указател*

```
char *str;
cin >> str;    // ERROR
(getline, …)
```

Низове
(като указател от тип char)

```
char str[7] = "123456";
*(str+1) = '9';
str[1] = '9';
cout << str;
```

Изход:
193456

```
const char *str2 = "123456";
*(str2+1) = '9';          // Error assignment of read-only location
str2[1] = '9';            // Error assignment of read-only location
cout << str2;
```

Задача

Дадена е квадратна матрица от низове.
Да се напише програмен фрагмент, който намира броя на палиндромите под главния диагонал (заедно с него).

```
// Вариант 1

int br = 0;

char a[MAX_SIZE][MAX_SIZE][MAX_SIZE2];

// Въвеждане на елементите на матрицата

for(int rowIndex = 0; rowIndex < MAX_SIZE; rowIndex++){
  for(int colIndex = 0; colIndex <= rowIndex; colIndex++) {

    // Дължина на низа
    int len = strlen(a[rowIndex][colIndex]);

    // Създаване на обърнат низ
    char revStr[MAX_SIZE2];
    for(int strIndex = len-1; strIndex >= 0; strIndex--) {
      revStr[len - strIndex - 1] = a[rowIndex][colIndex][strIndex];
    }

    revStr[len] = '\0';

    // Сравнение с обърнатия низ
    if (!strcmp(a[rowIndex][colIndex], revStr)) {
      br++;
    }
  }
}
```

Задача

Дадена е квадратна матрица от низове.
Да се напише програмен фрагмент, който намира броя на палиндромите под главния диагонал (заедно с него).

```
// Вариант 2

int br = 0;

char a[MAX_SIZE][MAX_SIZE][MAX_SIZE2];

// Въвеждане на елементите на матрицата

for(int rowIndex = 0; rowIndex < MAX_SIZE; rowIndex++){
  for(int colIndex = 0; colIndex <= rowIndex; colIndex++) {

    // Дължина на низа
    int len = strlen(a[rowIndex][colIndex]);

    // Директно сравнение
    …
  }
}
```