

УП Семинар 5

СИМВОЛНИ НИЗОВЕ

`std::string s;`

`char[] s;`

`char * s;`

`char c1, c2, c3;`



Що е то?

- Символният низ представлява последователност от символи.
- Казано тъй, че да ме разберете това е текст просто.
- Сега ще се научим как да боравим с него.

Как да дефинираме.

- Просто като масив от тип `char`
- В: Може ли в низ да се набути скандал на жена?

О: Миии засега не, защото не знаем колко ще е дълъг.

Пример

```
char str[100];
```

- В края на всеки низ се слага служебен символ, който указва края му. Този символ се нарича терминираща нула ('\0'). Затова винаги слагайте по един символ повече.

Сега как да въведем туй чудо

- Така?

```
#include <iostream>

int main()
{
    char str[100];
    int n;
    std::cin >> n;
    for(int i = 0; i < n; i++)
    {
        std::cin >> str[i];
    }
}
```

Ако искате 0 точки може би да.

- Ако не искате тогава така:

```
#include <iostream>

int main()
{
    char str[100];
    std::cin >> str;
}
```

cin.getline()

- Принципно със cin се въвежда низ, докато не се стигне до Space. Ако искаме да въведем цял ред използваме cin.getline(dest, size);

```
#include <iostream>
#include <cstring>

int main()
{
    char str1[101];
    std::cin.getline(str1, 100);
    std::cout << str1 << std::endl;
}
```

Сега как да работим с низове?

- Основните функции за работа със символни низове са: `strlen`, `strcpy`, `strcat`, `strcmp`, `strstr`
- Сега ще попитате ама къде е изтриването на подниз? Няма го!
- За да използваме тези функции, трябва да включим библиотека `cstring`.

Strlen

- Приема символен низ и извежда дължината му. Да не се бърка с размера!

```
#include <iostream>
#include <cstring>

int main()
{
    char str[] = "Hello";
    std::cout << strlen(str);
    //5
}
```

Strcpy

- Синтаксис: strcpy(dest, string)
- Копира string в dest.

```
#include <iostream>
#include <cstring>

int main()
{
    char str2[100];
    strcpy(str2, "Zdrasti");
    std::cout << str2;
}
```

Strcat

- Не котка, а конкатенация. Няма котенца тука.
- Синтаксис: `strcat(destination, source);`

```
#include <iostream>
#include <cstring>

int main()
{
    char str1[100];
    strcpy(str1, "Zdrasti ");
    char str2[] = "Gosho";
    strcat(str1, str2);
    std::cout << str1 << std::endl;
}
```

Как се сравняват низове?

- Така?

```
#include <iostream>
#include <cstring>

int main()
{
    char str1[100];
    strcpy(str1, "Zdrasti ");
    char str2[] = "Gosho";
    if(str1 == str2)
    {
        ///....
    }
}
```



**Straight to hell. To the boiler room of
hell. All the way down.**

Strcmp

- Сравнява низове лексикографски
- Синтаксис: `strcmp(str1, str2)`
- Връща:
 - 0 – ако са равни
 - >0 – ако `str1` е по-голям от `str2`
 - <0 – ако `str2` е по-голям от `str1`

Strstr

- Намира подниз.
- Засега това го приемайте на магия ако не знаете указатели. Ако пък знаете добре за вас.
- Синтаксис: `strstr(str, whatToFind)`
- Връща указател към първото срещане на `whatToFind`

```
#include <iostream>
#include <cstring>

int main()
{
    char str1[100];
    strcpy(str1, "Zdrasti ");
    if(strstr(str1, "drasti") != nullptr)
    {
        std::cout << "Found" << std::endl;
    }
}
```


Задачи

- Да се въведе низ с максимална дължина 100 и да се изведе дали е палиндром.
- Да се въведе низ и да се сумират числата от него.
- Да се въведе име и да се построи низ:
- Zdravei [Име]! Radvam se che izpolzvash moeto prilojenie.
- Въвеждат се низове до въвеждане на quit. Те да са от следния формат:
 - sum a b – извежда сумата на a и b
 - sub a b – извежда разликата на a и b
 - mult a b – извежда произведението на a и b
 - div a b – извежда частното на a и b
- Да се въведат n низа и да се изведе най-дългият от тях.
- Да се въведат n низа и да се изведе най-големият лексикографски от тях.