

Увод в програмирането

Лекция 8:

Работа с текст

Втора част – версия за **Информатика и Информационни системи**

Символен низ

- Слайдовете са за спец. **Инф. и ИС**
- Низ/стринг
- Представят се с масиви от `char`, завършващи със специален символ – с ASCII код 0 (`'\0'`)
- Литерали – **"Hello"** (използвали сме ги)
- Всичко от масиви важи
 - Някои действия обаче са по-лесни, например инициализация, отпечатване, въвеждане и др.

Представяне в паметта

- Масив от елементи от тип `char`
- Пример:
 - Низът **"ABC"** се представя по следния начин:

65	66	67	0
----	----	----	---

(65 е ASCII кодът на А, 66 – на В и т.н.)

- Последният символ е с ASCII код 0
 - Служи за индикация за край на низа – няма нужда да ползваме допълнителна променлива
 - Няма графично представяне
 - Литерал - `'\0'`

Деклариране и инициализиране

```
char string1[5];
```

// може да съдържа стрингове до 4 знака (един е за \0)

```
char string2[5] = "";
```

// това вече е празен низ, за разлика от горното

```
char string3[5] = "Hi!";
```

~~**string3 = "NO";**~~ // само при инициализация може

```
char string4[] = "Hello";
```

// заема 6 байта и не може да съдържа по-дълги низове

```
char string5[3] = {'H', 'i', '\0'};
```

// излишно сложно, а и можем да забравим нулата

~~**char string6[];**~~ //не може!

Дискусия

- Разлика между масив от `char` и символен низ
- Разлика между `'a'`, `"a"` и `a`, както и между `0`, `'0'` и `'\0'`

Вход/изход

- Отпечатване:

```
char string[100] = "Hello";  
cout << string << endl;
```

- Въпреки че string е масив, cout отпечатва Hello вместо адреса

- Прочитане:

```
cin >> string; //чете до първо срещане на  
интервал, нов ред или табулация
```

```
cin.getline(string, 100); //до 99 знака  
(включително интервали...), спира при Enter
```

Работа със СИМВОЛИТЕ на НИЗ

- Достъп до i-ти елемент на даден низ:
`char string[100] = "Hello";`
`char c = string[1]; // e`
- Обхождане на низ СИМВОЛ ПО СИМВОЛ:
- `for (int i = 0; string[i] != '\0'; i++)`
 `// работим със string[i]`

ИЛИ

```
for (int i = 0; string[i]; i++) ...
```

ИЛИ

```
for (char *c = string; *c; c++)  
    // работим със *c
```

И Т.Н.

Примери

- Нумерология
 - Лекторът не е фен, но ще направи компромис
- Хистограма – колко често се среща всеки СИМВОЛ
 - Тук вече лекторът е фен

Дефиниране на функции за работа с низове

- Отново важат правилата за масиви и отново нещата тук са по-лесни
- Подаване на низ като параметър:
достатъчен е само указател към началото
void f(const char *message)
void f(char *message)
- Това е така, понеже дължината се определя от позицията на символа \0
- Връщане на низ:
void f(char *result)
(подава се низ, в който да се запише резултатът)

Пример №1

- Проверка дали използването на скоби в даден израз е коректно
- Интересуват ни само скобите, без значение какво има около тях
- "" - да
- "2+(3*(5-" - не
- "()()())" - да
- "xyz)(abc" - не

Пример №2 – брой думи в текст

- bool isLetter(char c) { // засега поддържа само англ.
 return c >= 'a' && c <= 'z' || c >= 'A' && c <= 'Z';
}
int countWords(const char *string) {
 int result = 0;
 bool inWord = false;
 for (int i = 0; string[i]; i++) {
 if (!inWord && isLetter(string[i])) {
 // начало на нова дума
 inWord = true;
 result++;
 } else if (inWord && !isLetter(string[i])) {
 // край на дума
 inWord = false;
 }
 }
 return result;
}

Още примери

- Прочитане на естествено число, записано в низ (в 10-ична бройна система)
 - "123" → 123 (ако не сте били на лекциите и четете това въкъщи, няма да стане ясно какво се иска :))
 - Представяне в паметта: 49, 50, 51, 0 → 123
 - За съжаление не става с преобразуване с (int)
- Конструирание на символен низ, съдържащ 10-ичното представяне на естествено число
 - 123 → "123"
- Напишете ваши решения, но в реалния свят използвайте готовите функции `atoi` и `itoa` от `cstdlib` (`#include <cstdlib>`)

Малка дразнеща особеност

- Ако използваме `getline`, но преди това сме използвали `cin >> ...`, ще получим празен низ
- Слагаме `cin.ignore()`; преди `getline`
 - Или: дублираме извикването на `getline`
- Демонстрация

Още важни операции (1)

- За следващите примери ще ни трябва библиотеката `cstring`:
`#include <cstring>` или `<string.h>`
- Дължина на низ:
 - `strlen(string)` – брой символи на даден низ (нулата накрая не се брои!)
 - Грешно: `sizeof(string)`
- Присвояване на стойност:
 - Грешно: `str1 = str2;`
 - Вярно: `strcpy(str1, str2);`

Още важни операции (2)

- Лексикографско сравнение:
 - като подредбата в речника – напр.
 $a < аз < азбука < ах < боб$
 - **`strcmp(str1, str2)`**
 - Връща 0, ако стринговете са еднакви
`if (!strcmp(str1, str2)) cout << "equal";`
 - Връща число, по-малко от 0, ако $str1 < str2$
 - Грешно: $str1 == str2$ – така сравняваме адреси

Още важни операции (3)

- Конкатениране:
 - **strcat(dest, src)** – долепя src след края на dest
 - Резултатът се записва в dest
 - Грешно: `dest += src`
- Има още много други готови функции в `cstring`
 - Справка – чичо Google (или DuckDuckGo и т.н.)
- Демонстрация на функциите

Още демонстрации

- Масиви от низове
- Трикчета с указатели

Пример №1

- Функция, която добавя даден символ към края на даден низ
- ГРЕШНО решение от някои студенти от КН:

```
void addChar(char *string, char symbol) {  
    string[strlen(string)] = symbol;  
    string[strlen(string) + 1] = '\0';  
}
```
- Решението е грешно в общия случай. В някои ситуации може и да работи
- Защо е грешно?!

Пример №2

- Премахване на интервали в началото и в края на низ

```
void trim(char *string) {  
    int len = strlen(string);  
    if (len > 0) { // не забравяме да проверяваме всякакви случаи!  
        // премахване на интервали в края:  
        int last = len - 1;  
        while (last >= 0 && string[last] == ' ' )  
            last--;  
        string[last + 1] = '\0';  
        // премахване на интервали в началото:  
        int first = 0;  
        while (string[first] == ' ' )  
            first++;  
        if (first > 0)  
            strcpy(string, string + first);  
        // или ако горният ред ни изглежда труден -  
        // пускаме един for цикъл и трием първите елементи  
    }  
}
```

- Това беше всичко за днес