Семинар по „Увод в програмирането“

Низове

1. Какво е низ?

Низът е поредица от символи. В себе си той може да съдържа главни и малки букви, цифри и символи.В този курс низовете ще ги представяме като масив от символи(**char array**). Щом ползваме масив от **char-ове**. Зад всеки символ съответства число, което можем да видим в **ASCII** таблицата. Всеки низ е с точно определен размер като винаги последния символ е **'\0'**(терминираща нула), указваща края на символния низ.

```
//static
char text[15];
//dynamic
char* text = new char[15];
```

2. Въвеждане и извеждане на низ

Има два начина да въведем низ от конзолата.

Единият е със **std::cin.** По този начин можем да четем символи от конзолата до първото разстояние(whitespace), който срещнем.

```
char text[16];
std::cin >> text;

std::cout << text;
```

Резултат:

```
I like to eat pasta.
I
```

Другият начин да четем низ от конзолата е използвайки функцията

std::cin.getline(char* text, streamsize n, char delimiter). Чрез **getline** можем да четем символи от конзолата или докато прочетем **n** на брой символа, или докато прочетения символ не е равен на **delimiter. Delimiter** може да се изпусне като по подразбиране символът ще бъде **'\n'**.

```
char text[16];
std::cin.getline(text, 16);

std::cout << text;
```

Резултат:

```
This is dog!
This is dog!
```

За да се избегне грешно прочитане на данни, останали в потока, които могат да бъдат прочетени чрез **std::cin.getline()**, е добра практика потока да се изчиства предварително чрез **std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');.** Това е необходимо особено, когато преди това има въвеждане на данни от конзолата чрез **std::cin.** Вместо **std::numeric_limits<std::streamsize>::max()** е достатъчно аргумента да е достатъчно голямо число. Например 256.

Пример, в който не изтрива потока от излишни данни:

```
char text[16];
std::cin >> text;
std::cout << "String read with cin: " << text << std::endl;

char secondText[16];
//std::cin.ignore(256, '\n');
std::cin.getline(secondText, 16);
std::cout << "String read with cin.getline: " << secondText << std::endl;
```

И съответния резултат:

```
This is dog!
String read with cin: This
String read with cin.getline:  is dog!
```

Пример, в който изтриваме потока от излишни данни:

```
char text[16];
std::cin >> text;
std::cout << "String read with cin: " << text << std::endl;

char secondText[16];
std::cin.ignore(256, '\n');
std::cin.getline(secondText, 16);
std::cout << "String read with cin.getline: " << secondText << std::endl;
```

Резултат:

```
This is dog!
String read with cin: This
This is dog!
String read with cin.getline: This is dog!
```

Низ извеждаме със **std::cout.**

3. Достъп и обхождане на низ

Достъпваме отделните символи от низ така, както достъпваме елементи на масив, чрез индекс.

```
char text[15] = “This is dog!”;
char randomSymbol = text[0];//’T’
```

Низ обождаме с помощта на цикъл:

```
char text[16];
std::cin >> text;

for (int i = 0; text[i] != '\0'; i++)
{
    text[i] += 1;
    std::cout << text[i];
}
```

Резултат:

```
ABC
BCD
```

4. Някои готови функции

Следните функции са от библиотеката **<string>:**

1) strlen()-връща дължината на низ без ‘\0’
2) strcpy(str1, str2)-копира низа str2 в str1
3) strcmp(str1, str2)-прави лексикографско сравнение на двата низа
4) strcat(dest, str)-конкатенира низа dest с низа str

atoi() е от библиотеката <**stdlib**> и тя конвертира низ в число.

Както знаем тези функции не са позволени да се използват наготово в рамките на този курс, но тук са изложени, за да може читателя да добие представа за някои функции от библиотеката **<string>.**