Структурен тип данни – символен низ

1. **Деклариране на символен низ**
   a. **Определение –** последователност от краен брой елементи от символен тип, оградени с кавички и завършващи със знак '\0'
   b. **Синтаксис –** на практика символния низ е масив от елементи с базов тип char

char <променлива>[<дължина>];

char – запазена дума
< променлива > - име на низа, идентификатор
[< дължина >] – константа от изброен тип, който задава дължината на низа (стойност по-голяма от дължината с единица от максималния брой символи в низа)

   c. Семантика – Заделя се в ОП <дължина> на брой последователни клетки, всяка с големина 1B. Първите n-1 клетки са определени за елементите на низа, а последната е служебна за символа указващ край на низа'\0'
   d. Примери:

След дефиницията
char s[4];
char w[10]="abba";
разпределението на паметта има вида:
ОП

s	s[0]	s[1]	s[2]	s[3]			
	-	-	-	-			
w	w[0]	w[1]	w[2]	w[3]	w[4]	w[5]	.....w[9]
	97	98	98	97	\0	\0	\0

char str[10];// в ОП се заделят 10 последователни клетки по 1B, последната клетка е '\0'
char family[15];// в ОП се заделят 15 последователни клетки по 1B, последната клетка е '\0'

2. **Инициализиране на символен низ – <u>само на реда, в който е декларирана</u>**

<u>Първи начин:</u>

char name[5]={'I','v','a','n'};// в ОП → I v a n \0
0 1 2 3 4

char name[9]={'I','v','a','n'};// в ОП I v a n \0 \0 \0 \0 \0
0 1 2 3 4 5 6 7 8

Втори начин:
char name[5]=”Ivan”; //задаване символите в кавички

Трети начин
char name[]=”Ivan”; //без явно задаване на дължината на низа и се определя от броя на символите, зададени за инициализиране и символа за край на низ

Недопустими инициализации:
char name[5];
name={‘I’,’v’,’a’,’n’};
или
char name[5];
name=”Ivan”;

3. **Въвеждане и извеждане на низ**
   a. **Въвеждане на низ от клавиатурата:**

Синтаксис:

cin>><променлива>;

Семантика:
Въвежда от клавиатурата символи до натискане на клавиш за нов ред, интервал, табулация или до запълване на целия низ. Знакът за край на низ се въвежда автоматично от системата при натискане на горепосочените клавиши.
Пример:
char name[20];
cin>>name;

   b. **Чрез функцията getline**

Синтаксис

cin.getline(<променлива>,<size>,[<char>]$_{опц}$)

<променлива> е име на низ, идентификатор;
<size> е цял израз
<char> е произволен символ
Ако <char> е пропуснато, подразбира се исимволът \n

Семантика
Въвежда от буфера на клавиатурата редица от символи с максимална дължина <size>-1. Въвеждането продължава до срещане на символа, зададен в <char> или до

въвеждане на <size>-1 символа (ако междувременно не е достигнат символът от <char>)

Примери:

char s1[100];

cin.getline(s1,10);

Настъпва пауза. Очаква се въвеждането на низ, след което да се натисне клавиша ENTER. Ако дължината на въведения низ е по-голяма от 9 преди да е натиснат клавишът ENTER, вземат се първите 9 символа, свързват се с променливата s1 и изпълнението завършва. В противен случай, в s1 се записва редицата от символи без символа \n.

char s2[200];

cin.getline(s2,200,'.');

Настъпва пауза. Очаква се въвеждането на низ, след което да се натисне клавиша ENTER. Ако дължината на въведения низ е по-голяма от 199 преди да е въведен символът '.', вземат се първите 199 символа, свързват се с променливата s2 и изпълнението завършва. В противен случай, в s2 се записва редицата от символи до символа '.' (без него).

Извеждане на символ на монитора:

Синтаксис:

Cout<<<променлива>;

Семантика:

Извежда на екрана низа с име <променлива>

Пример:

char name[20]="Ivan";

cout<<name;

4. **Достъп до отдалечен елемент на низ –** чрез неговия индекс и номерирането започва от 0 (аналогично на достъп до елемент на масив)