Низове

Стринговете в същността си представляват последователност от символи, в този урок ще ги разгледаме като масив от тип **char**.

Единствената особеност при тази репрезентация на стринговете е, че последният символ от нашия низ трябва да бъде ‘\0’.

 Това е специален символ, обозначаващ края на нашия низ. Това е необходимо, тъй като големината на низовете е статична (за разлика от големината на стринговете, която е динамична).

Например, ако декларираме символен масив от 15 елемента, но решим да съхраним думата „Pineapple“ в него, ние ще запълним само първите 9 елемента от масива. И тъй като последните 6 елемента са неопределени, трябва да се избегне достъпа им. Точно поради тази причина, непосредствено след последния определен елемент от масива, трябва да се постави символа ‘\0’, за да се обозначи края на низа и да не се обхожда по-натам. Трябва също така да внимаваме да не превишаваме ограничението от 15-те символа (в случая). Нагледна репрезентация:

Забележете, че ако декларираме низ по следния начин:

1. char myCString [] = „Pineapple“;

символът ‘\0’ се поставя автоматично в края на низа. Затова и тази форма на декларация е за предпочитане пред стандартната за масиви, тъй като при нея трябва да добавим ръчно ‘\0’ в края:

1. char myCString [] = { ‘P’, ‘i’, ‘n’, ‘e’, ‘a’, ‘p’, ‘p’, ‘l’, ‘e’, ‘\0’ };

И в двата случая масивът myCString е деклариран с дължина 10 елемента, 9 символа за думата „Pineapple“ и 1 за последния знак.

Обърнете внимание, че говорим за инициализиране на масив от символи в момента на неговото деклариране, а не за присвояване на стойности след неговото деклариране. Не трябва да забравяме, че това са преди всичко масиви и сме зависими от техните ограничения. Тоест след началното инициализиране на масива, НЕ е възможно просто да му присвояваме стойности по следния начин:

1. myCString = „Banana“;

Присвояването на стойности трябва да се прави поотделно за всеки елемент от масива:

1. myCString[0] = ‘P’;
2. myCString[1] = ‘e’;
3. myCString[2] = ‘a’;
4. myCString[3] = ‘r’;
5. myCString[4] = ‘\0’;

Ето един пример, който принтира низове:

* #include <stdio.h>
* int main () {
* char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
* printf("Greeting message: %s\n", greeting );
* return 0;
* }
* Резултат:−
* Greeting message: Hello

Забележете, че конвертирането на низ в стринг става бързо и лесно:

1. char myCString[] = „Pineapple“;
2. string myString = myCString;

Включвайки header файла <string>, можем за използваме C++ функции от стандартната библиотека, работещи със стрингове и низове. Някои от по-използваните са:

* strcpy (str2,str1) – копира str1 във str2
* strcat (str2,str1) – добавя str1 в края на str2
* strcmp (str1,str2) – сравнява str1 със str2
* strlen (str) – връща дължината на стринга/низа
* [Live Demo](http://tpcg.io/P0muN5)
* C supports a wide range of functions that manipulate null-terminated strings −

|  |  |
| --- | --- |
| **Sr.No.** | **Function & Purpose** |
| 1 | **strcpy(s1, s2);*** копира str1 във str2
 |
| 2 | **strcat(s1, s2);*** добавя str1 в края на str2
 |
| 3 | **strlen(s1);*** връща дължината на стринга/низа
 |
| 4 | * **strcmp(s1, s2);** сравнява str1 със str2

Връща 0 if s1 == s2 , отрицателно число if s1<s2; положително число if s1>s2. |
|  |  |
|  |  |

* Пример:
* Важно е да включите библиотеката <string.h>, за да имате достъп дп фукнциите
* [Live Demo](http://tpcg.io/VBA8Qx)
* #include <stdio.h>
* #include <string.h>
* int main () {
* char str1[12] = "Hello";
* char str2[12] = "World";
* char str3[12];
* int len ;
* /\* copy str1 into str3 \*/
* strcpy(str3, str1);
* printf("strcpy( str3, str1) : %s\n", str3 );
* /\* concatenates str1 and str2 \*/
* strcat( str1, str2);
* printf("strcat( str1, str2): %s\n", str1 );
* /\* total lenghth of str1 after concatenation \*/
* len = strlen(str1);
* printf("strlen(str1) : %d\n", len );
* return 0;
* }
* Резултат:
* −
* strcpy( str3, str1) : Hello
* strcat( str1, str2): HelloWorld
* strlen(str1) : 10

Зaдължителни задачи:

**Задача 1**Да се състави програма чрез която от клавиатурата се въвеждат 30-50 знака общо.
Като резултат на екрана да се извежда броя знаци, 1-вия, средния и последния знак от въведения текст.
**Пример**: abcdefgh
**Изход**: първи a, среден e последен h

Решение:

1.#include <string> //strupr, strlwr,strlen

02.#include <stdio.h>

05.using namespace std;

06.int main()

07.{ char mas[80];//deklarirane na promenliwite

08.int dyl,a,b,c;

09.printf(“Systawete programa, chrez koqto ot klawiaturata se wywevdat\n)";

10. printf(“30-50 znaka obsho. Kato rezultat na ekrana da se izwevda\n");

11. printf(“broq znaci, 1-wiq, sredniq i posledniq znak ot wywedeiq tekst.\n");

12. printf("Primer: abcdefgh Izhod pyrwi a, sreden e posledn h");

13. printf(“Wywedete tekst: ";

  scanf(“%s”, &mas);//wywevda niza

14.dyl=strlen(mas);//opredelq dylvinata na wywedeniq tekst

15.a=dyl/2;//nomera na sredniq znak

16.printf("Dylvinata na wywedeniq tekst e:%d ",dyl);

17.ptintf("Pyrwiqt simwol ot teksta e: %c”,mas[0]);

18.printf("Sredniqt simwol ot teksta e: %c”, mas[a]);

19.printf("Posledniqt simwol ot teksta e: %c”,mas[dyl-1]);

20.

21.return 0;

22.}

**Задача 2: Да се намери пълното име name при зададени малко име firstName и фамилия familyName:**

#include < iostream.h >

#include < string.h >

int main(){

 char firstName[20]="Ivan", familyName[20]="Petrov";

 char name[20]="";

 strcat(name,firstName);

 strcat(name," ");

 strcat(name,familyName);

 cout<< name<< endl;

return 0;

}

\*заменете Ivan I Petrov с четене от клавиатурата

**Задача 3** Да се състави програма , чрез която от клавиатурата се въвеждат последователно две думи с дължина 10-20 знака.

Програмата да размени първите им 5 знака и да изведе дължината на по-дългата, както и новото им съдържание.

Пример: uchilishe uchenik

Изход: 9 uchenishe

**Задача 3** Да се състави програма на C, чрез която от клавиатурата се въвежда низ и знак.

Като резултат на екрана се извежда съобщение дали знакът се съдържа в низа и се отпечатват всички следващи го знаци от низа.

Пример: Uchitel; i

Изход: Знакът 'i' е на позиция 4, След него е 'itel'.

**Задача 4**: Напишете функция, която за даден знаков низ връща като резултат броя на цифрите в него.

**Задача 5**: Да се напише функция, която сравнява лексикографски два низа s1 и s2, и връща 0 ако са равни, -1 ако s1<s2, и 1 ако s1>s2. Да не се използва функцията int strcmp(const char\*, const char \*).

Допълнителни задачи -състезателни с решение:

**Машина на времето (Задача D6 - НОИ-3, Плевен, 2010 г.)**

**Изобретателят Айн Щайн създал машина на времето. Машината пътувала в кръг, като по часовник.  Tръгвала от настоящата 2010 година и пътувала напред в бъдещето до 4020 година. Щом я достигнела се пренасяла в година 1005 след Христа и пак тръгвала напред към настоящето.
Изобретателят Щайн разрешавал пътуване само ако желаещият успее да реши задача и отговори на въпроса в коя точно година ще го пренесе машината. Желаещите получават текст до 18 символа, в който има поне една цифра, различна от нула. Отделят се само цифрите (k на брой)  и с тях се образуват най-голямото и най-малкото възможни k-цифрени числа, като всяка цифра участва точно един път.  Разликата на тези две числа показва колко години ще пътува машината, тръгвайки от 2010 и пътувайки в кръга.
Помогнете на малкия изследовател Все Знайко, като създадете програма machine, която извежда точната година, в която машината ще се пренесе.**

 **Ограничения: 1<=k<=18**

 **Вход: На стандартния вход се въвежда текст до 18 символа.
Изход: На стандартния изход се извежда годината, в която машината ще се пренесе.**

**Пример:**

|  |  |
| --- | --- |
| **вход** | **изход** |
| **TiMe312MacHiNe** | **2208** |

**Задачата може да бъде решена по следния начин:
Отделят се цифрите от низа в масив и се сортират. Генерират се най-голямото и най-малкото число. При генерирането на най-малкото число трябва да се съобрази, че за да бъде то k – цифрено не трябва да има водещи нули. Намира се разликата им и се изчислява в коя година ще попадне машината при движението си в описания кръг.**

**Програмна реализация:
#include<iostream>
#include<string>
using namespace std;**

**int main()
{
char niz[20];
int i,j,swap,n,k=0,a[20],p;
unsigned long long A=0,B=0,C;
cin.getline(niz,20);
n=strlen(niz);**

**for(i=0;i<n;i++)
{
if(niz[i]>=48 && niz[i]<=57)
{
a[k]=niz[i]-48;
k++;
}
}
for(i=0;i<k-1;i++)
for(j=0;j<k-i-1;j++)
if(a[j]<a[j+1])
{
swap=a[j];
a[j]=a[j+1];
a[j+1]=swap;
}**

**for(i=0;i<k;i++)
A=A\*10+a[i];

p=k-1;
if(a[p]==0)
{**

**while(a[p]==0&&p>0) p--;
a[k-1]=a[p];
a[p]=0;
}
for(i=0;i<k;i++)
B=B\*10+a[k-i-1];**

**C=((A-B)+1005)%3015+1005;
cout<<C<<endl;
return 0;
}**

**ДУМИ (Задача В1, НОИ-1, 2010 г.)**

**Всяка дума има свое тегло. То се определя от номерата на буквите в думата и от тяхната позиция.
Букви  A и a имат номер 1, букви B и b  – номер 2, букви C и c –  номер 3, … , букви Z и z  имат номер 26.
Теглото на думата се определя по следния начин  – номерът на първата буква се умножава по едно,  на втората – по две, на третата – по 4, на четвъртата – по 8, на петата – по 16  и т.н. Сборът на тези произведения  определя колко „тежи” дадената дума.. Например теглото на думата Kod се изчислява така:
11 x 1 + 15 x 2 + 4 x 4 = 57.

Напишете програма dumi, която за дадено множество от думи намира най-тежката дума.**

**Вход: От първия ред на стандартния вход се въвежда броя на думите *n*, а от следващите *n* реда самите думи.

            Изход: На стандартния изход да се изведе „най-тежката” дума. Ако има няколко „най-тежки” думи, да се изведе първата по реда на въвеждане.**

**Ограничения
2 <  *n*< 100
Всяка от дадените думи има най-много 20  букви.**

**Пример:**

|  |  |
| --- | --- |
| **вход** | **изход** |
| **3       KoddEnQweRtY** | **QweRtY** |