

Какие фрагменты программы дадут одинаковые результаты?

<pre>//1 s=0; for (i=0;i<n;i++) for(j=0;j<i; j++) s+=a[i][j]; cout<<s<<"\n"; //2 s=0; for (i=0;i<n;i++) for(j=0;j<i; j++) s+=a[j][i]; cout<<s<<"\n";</pre>	<pre>//3 s=0; for (i=n-1;i>-1;i--) for(j=0;j<i; j++) s+=a[i][j]; cout<<s<<"\n"; //4 s=0; for (i=0;i<n;i++) for(j=n-1;j>-1; j--) s+=a[j][i]; cout<<s<<"\n";</pre>
--	--

C:\WINDOWS\system32\cmd.exe

```
4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
10
20
10
40
Для продолжения нажмите любую клавишу . . .
```

Символьная информация и строки

Лекция 11

Символьный тип данных

Базовый тип данных **char** :

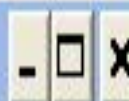
- отводится 1 байт памяти;
- целое со знаком (в диапазоне $-127 \dots +127$) ;
- как **СИМВОЛ текста**.

Тип `char`

- не имеет никаких ограничений на выполнение операций, допустимых для целых переменных: от операций сравнения и присваивания до арифметических операций и операций с отдельными разрядами.

```
void main()
{
char s, c;
for (s='A'; s <= 'Z'; s++)
{
    if (s%10==0) printf("\n");
    printf("%c %d\t",s,s);
}
printf("\n");
for (c=0x41; c <=0x5A; c++)
{
    if (c%10==0) printf("\n");
    printf("%c %d\t",c,c);
}
}
```

C:\WINDOWS\system32\cmd.exe



A	65	B	66	C	67	D	68	E	69										
F	70	G	71	H	72	I	73	J	74	K	75	L	76	M	77	N	78	O	79
P	80	Q	81	R	82	S	83	T	84	U	85	V	86	W	87	X	88	Y	89
Z	90																		
A	65	B	66	C	67	D	68	E	69										
F	70	G	71	H	72	I	73	J	74	K	75	L	76	M	77	N	78	O	79
P	80	Q	81	R	82	S	83	T	84	U	85	V	86	W	87	X	88	Y	89
Z	90																		

Для продолжения нажмите любую клавишу . . .


```
void main()
```

```
{
```

```
int n;
```

```
char c;
```

```
cin>>n;
```

```
c = n + '0';
```

```
printf("%c\n",c);
```

```
if (n <=9) c = n + '0'; else c = n - 10 + 'A';
```

```
printf("%c\n",c);
```

```
}
```

C:\WINDOWS\system32\cmd.exe

8
8
8
Для продолжения нажмите любую клавишу . . .



C:\WINDOWS\system32\cmd.exe

11

;

В

Для продолжения нажмите любую клавишу . . .

- Получить значение **целой переменной** из **символа десятичной цифры**:

```
if (c >='0' && c <='9') n = c - '0';
```

- Получить значение **целой переменной** из **шестнадцатеричной цифры**:

```
if (c >='0' && c <='9') n = c - '0';
```

```
    else
```

```
if (c >='A' && c <='F') c = c - 'A' + 10;
```

- Преобразовать **маленькую латинскую букву** в **большую**:

```
if (c >='a' && c <='z') c = c - 'a' + 'A';
```

Представление символьной информации

- Для представления символьной информации используются
 - символы,
 - символьные переменные,
 - текстовые константы.

- символ занимает один байт, его значение не меняется

```
const char c='c';
```

- символьные переменные, занимают по одному байту, значения могут меняться

```
char a,b;
```

- текстовая константа

```
const char *s="Пример строки";
```

- **Общий вид описания переменных строкового типа:**

`char имя_массива[кол-во символов в строке];`

`char имя_массива[];`

`char *имя_массива;`

Строки в стиле C

- Строка в C – это массив символов, заканчивающийся нуль-символом – '\0' (нуль-терминатором).
- По положению нуль-терминатора определяется фактическая длина строки.
- Количество элементов в таком массиве на 1 больше, чем изображение строки.

A	\0
"A" строка (2 байта)	

A
'A' символ (1 байт)

- Присвоить значение строке с помощью оператора присваивания нельзя.
- Поместить строку в массив можно либо при вводе, либо с помощью инициализации.

Пример 1

```
#include <iostream>
#include <math.h>
using namespace std;
void main()
{
    char s1[10]="string1";
    int k=sizeof(s1);
    cout<<s1<<"\t"<<k<<endl;
    char s2[]="string2";
    k=sizeof(s2);
    cout<<s2<<"\t"<<k<<endl;
    char s3[]={ 's','t','r','i','n','g','3','\0'};
    k=sizeof(s3);
    cout<<s3<<"\t"<<k<<endl;
    //указатель на строку, ее нельзя изменить:
    char *s4="string4";
    k=sizeof(s4);
    cout<<s4<<"\t"<<k<<endl;
}
```

C:\WINDOWS\system32\cmd.exe

string1 10

string2 8

string3 8

string4 4

Для продолжения нажмите любую клавишу . . .

Пример 2

```
char *s="String5";    //выделяется 8 байтов для строки
```

```
char* ss;             //описан указатель  
ss="String6";         // ОШИБКА – не выделена память
```

```
char *sss=new char[10]; //выделяем динамическую память  
strcpy(sss,"String7");  //копируем строку в память
```

Ввод-вывод строк в стиле C

- `int getchar(void)` – осуществляет ввод одного символа из входного потока, при этом она возвращает один байт информации (символ) в виде значения типа `int`.
- `int putchar (int c)` – помещает в стандартный выходной поток символ `c`.
- `char* gets(char*s)` – считывает строку `s` из стандартного потока до появления символа `'\n'`, сам символ `'\n'` в строку не заносится.
- `int puts(const char* s)` записывает строку в стандартный поток, добавляя в конец строки символ `'\n'`, в случае удачного завершения возвращает значение больше или равное 0 и отрицательное значение (`EOF=-1`) в случае ошибки.

```
#include <stdio.h>
void main()
{ char ch;
  while((ch=getchar()) !='\n') putchar(ch);
  putchar(ch);

}
```



C:\WINDOWS\system32\cmd.exe

мир спорт 2015

мир спорт 2015

Для продолжения нажмите любую клавишу . . .

Примеры

Пример 1

```
char s[20];  
cin>>s;      //ввод строки из стандартного потока  
cout<<s;     //вывод строки в стандартный поток
```

Пример 2

```
char s[20];  
gets(s);     //ввод строки из стандартного потока  
puts(s);     //вывод строки в стандартный поток
```

Пример 3

```
char s[20];  
scanf("%s",&s); //ввод строки из стандартного потока  
printf("%s",s); //вывод строки в стандартный поток
```



```
include <stdio.h>
void main()
{ char a[20];
scanf("%s",&a);
printf("%s",a);
printf("\n");
}
```



C:\WINDOWS\system32\cmd.exe

мир спорт 2015

мир

Для продолжения нажмите любую клавишу . . .

```
#include <iostream>
using namespace std;
void main()
{ char a[20];
cin>>a;
cout<<a;
cout<<"\n";
}
```

C:\WINDOWS\system32\cmd.exe

мир спорт 2015

мир

Для продолжения нажмите любую клавишу . . .

```
#include <stdio.h>
void main()
{ char a[20];
  gets(a);
  puts(a);
  printf("\n");
}
```



C:\WINDOWS\system32\cmd.exe

мир спорт 2015

мир спорт 2015

Для продолжения нажмите любую клавишу . . .

```
for (i=0; B[i] !='\0'; i++)...
```

Или можно увеличивать указатель на 1,
пока очередным символом не станет
нуль:

```
while (*st++ ) { ... }
```

```
int str_len(char *st)
{ char* p=st;  // чтобы не портить
               указатель (может оказаться за
               пределами строки)
  int len = 0;
  while ( *p++ ) ++len;
  return len;
}
```


Определение текущей длины строки:

```
unsigned int strlen(char *S)  
{ char *S0=S;  
  while (*S) S++;  
  return(S-S0);  
}
```

Копирование строк:

```
char *strcpy(char *d, char *S) // d – куда  
    копируем,  
    // S – что копируем  
{ char *r=d;  
    while(*S) {*d=*S;S++;d++;}  
    *d='\0';  
    return r;  
}
```

- Для работы со строками все действия реализуются через стандартные функции, которые находятся в библиотеке “**string.h**” или **<cstring>**

Библиотечные функции для работы со строками

Прототип функции	Краткое описание	Примечание
<code>unsigned strlen (const char* s);</code>	Вычисляет длину строки s.	
<code>int strcmp (const char* s1, const char* s2);</code>	Сравнивает строки s1 и s2.	Если $s1 < s2$, то результат отрицательный, если $s1 == s2$, то результат равен 0, если $s1 > s2$ – результат положительный.
<code>int strncmp (const char* s1, const char* s2, int n);</code>	Сравнивает первые n символов строк s1 и s2.	Если $s1 < s2$, то результат отрицательный, если $s1 == s2$, то результат равен 0, если $s1 > s2$ – результат положительный.

Прототип функции	Краткое описание	Примечание
<code>char* strcpy(char* s1, const char* s2);</code>	Копирует символы строки s2 в строку s1.	
<code>char* strncpy(char* s1, const char* s2, int n);</code>	Копирует n символов строки s2 в строку s1	Конец строки отбрасывается или дополняется пробелами.
<code>char* strcat(char* s1, const char* s2);</code>	Приписывает строку s2 к строке s1	
<code>char* strncat(char* s1, const char* s2, int n);</code>	Приписывает первые n символов строки s2 к строке s1	

Прототип функции	Краткое описание	Примечание
<pre>char *strchr(const char *s, int c);</pre>	<p>возвращает указатель на первое вхождение символа c в строку, на которую указывает s. Если символ c не найден, возвращается NULL</p>	
<pre>long atol(const char *s)</pre>	<p>преобразует строку в длинное целое число, в случае неудачного преобразования возвращается 0</p>	

```
include <iostream>
using namespace std;
int words(char c[])  //--- Подсчет количества слов
{ int i,nc;
  for (nc=0,i=0;c[i]!='\0';i++)
  {  // Посимвольный просмотр строки
    if (c[i]!=' ' && (i==0 || c[i-1]!=' ')) nc++;
  }
  return nc;
}
void main()
{ char s[80];
  gets(s);
  cout<<words(s)<<"\n";
}
```

C:\ C:\WINDOWS\system32\cmd.exe

С наступающим Новым 2015 годом!

5

Для продолжения нажмите любую клавишу . . .

C:\WINDOWS\system32\cmd.exe

: С наступающим Новым 2015 годом!

5

Для продолжения нажмите любую клавишу . . .

```
#include <iostream>
using namespace std;
int words(char c[])    //--- Подсчет количества слов
{ int i,nc;
  nc=0;i=0;
  while (c[i]!='\0')
  {
    while(c[i]==' ')i++; //пропуск пробелов
    while(c[i]!=' ' && c[i]!='\0' )i++;
    nc++;
    if (c[i]!='\0') i++;
  }

  return nc;
}

void main()
{ char s[80];
  gets(s);
  cout<<words(s)<<"\n";
}
```

C:\ C:\WINDOWS\system32\cmd.exe

С наступающим Новым 2015 годом!

5
Для продолжения нажмите любую клавишу . . .

```
#include <stdio.h>
#include <string.h>
#include <iostream>
using namespace std;
void main()
{
char s[150], //исходная строка
w[25],      //слово
mas[10][25]; //массив слов
int k=0,t=0,i,len,j;

puts("\nInput stroky");
gets(s);
puts(s);
//puts("\n");
len=strlen(s);
cout<<len<<"\n";
```

```
while(t<len)
{
for(j=0,i=t;s[i]!=' '&&
s[i]!='\0';i++,j++)
w[j]=s[i];    //формируем слово
до пробела
w[j]='\0';//формируем конец
строки
puts(w);
strcpy(mas[k],w); //копируем
слово в массив
k++;            //увеличиваем
счетчик слов

//переходим к следующему
слову в исходной строке
t=i+1;
}
```

```

cout<<"k="<<k<<"\n";
strcpy(s, " ");//очищаем исходную
строку
for(t=0;t<k;t++)
{   puts(mas[t]);
if(mas[t][0]<'0' || mas[t][0]>'9')
//если первый символ не цифра
{strcat(s,mas[t]);//копируем в
строку слово
strcat(s, " ");//копируем в строку
пробел
puts(s);
}
}
puts ("ОТВЕТ\n");
puts(s);//выводим результат
}

```

```

C:\WINDOWS\system32\cmd.exe
Input stroky
мир спорт 123 май 567 лето
мир спорт 123 май 567 лето
26
мир
спорт
123
май
567
лето
k=6
мир
мир
спорт
мир спорт
123
май
мир спорт май
567
лето
мир спорт май лето
ОТВЕТ

мир спорт май лето
Для продолжения нажмите любую клавишу . . .

```

Дана строка символов. Подсчитать,
сколько различных символов
встречается в ней. Вывести их на экран.

Введите строку: **11223344**

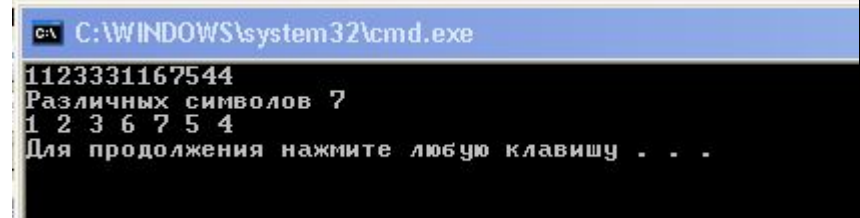
Различных символов: 4

1 2 3 4

```

#include <locale.h>
using namespace std;
void main()
{ int i,j=0;
  setlocale(LC_ALL,"rus");
  char s[50],diff[50]="\0";
  gets(s);
  for(i=0;s[i]!='\0';i++)
    if (strchr(diff,s[i])==NULL) // Если
    символ в строке не найден, то NULL
    {
      diff[j]=s[i];
      j+=1;
      diff[j]='\0';
    }
  cout << "Различных символов " <<
  strlen(diff)<<endl;
  for (i=0;diff[i]!='\0';i++)
    cout<<diff[i]<<" ";
  // cout<<diff; Выведет без пробелов
  cout<<endl;
}

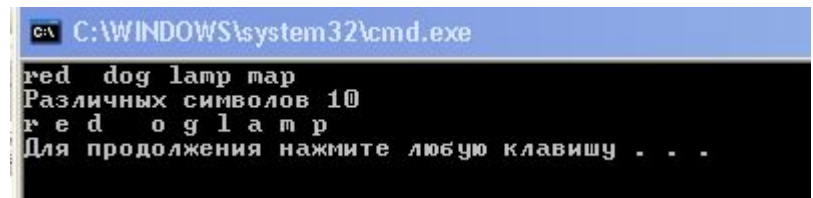
```



```

C:\WINDOWS\system32\cmd.exe
1123331167544
Различных символов 7
1 2 3 6 7 5 4
Для продолжения нажмите любую клавишу . . .

```



```

C:\WINDOWS\system32\cmd.exe
red dog lamp map
Различных символов 10
r e d   o g l a m p
Для продолжения нажмите любую клавишу . . .

```

Строки в стиле C++

- C++ строки определены в библиотеке `<string>`, которую требуется подключить с помощью директивы

```
#include <string>
```


Что будет выведено на экран?

```
#include <iostream>
#include <string>
using namespace std;

void main()
{
    char s1[20],s2[20];
    int f;
    strcpy(s1,"ПРИВЕТ СТРАНА");
    strcpy(s2,"СТРАНА ПРИВЕТ");
    f=strcmp(s1,s2);

    if(f>0) cout<<"s1>s2"<<endl;
    else
        if(f<0)cout<<"s1<s2"<<endl;
        else cout<<"s1==s2"<<endl;
}
```




C:\WINDOWS\system32\cmd.exe

s1<s2

Для продолжения нажмите любую клавишу . . .

СТРОКИ В СТИЛЕ C++

Лекция 12

Создание строк в стиле C++

- `string s;`//пустая строка
- `string s(cstr);`//создает строку из C строки
- `string s(cstr, len);`// создает строку из len
//символов C строки
- `string s(num, ch);`// создает строку из num //
символов ch
- `string s(str);`// создает строку из строки str

```
#include <iostream>
#include <locale.h>
#include <string>
using namespace std;

void main()
{
    string s1;
    char str1[ ]="old string";

    string s2(str1);
    cout<<"s2="<<s2<<endl;

    string s3(str1,3);
    cout<<"s3="<<s3<<endl;
```

```
    string s4(5,'5');
    cout<<"s4="<<s4<<endl;

    string s5("new string");
    cout<<"s5="<<s5<<endl;

    string s6(s5);
    cout<<"s6="<<s6<<endl;
}
```

C:\WINDOWS\system32\cmd.exe

s2=old string

s3=old

s4=55555

s5=new string

s6=new string

Для продолжения нажмите любую клавишу . . .

Операции

=	присваивание	>	больше
+	конкатенация	>=	больше или равно
==	равенство	[]	индексация
!=	неравенство	<<	вывод
<	меньше	>>	ввод
<=	меньше или равно	+=	добавление


```
#include <iostream>
#include <locale.h>
#include <string>
using namespace std;

void main()
{
    string s1;
    char str1[ ]="old string";
    string s2(str1);
    cout<<"s2="<<s2<<endl;
    s1=s2;
    cout<<"s1="<<s1<<endl;
    string s3(str1,3);
    cout<<"s3="<<s3<<endl;
    string s4(5,'5');
    cout<<"s4="<<s4<<endl;
```

```
    cout<<"s3>s4 - "<<(s3>s4)<<endl;
    cout<<"s3<s4 - "<<(s3<s4)<<endl;
    string s5("new string");
    cout<<"s5="<<s5<<endl;
    string s6(s5);
    cout<<"s6="<<s6<<endl;
    cout<<"s5==s6 - "<<(s5==s6)<<endl;
    cout<<"s5+s4 - "<<(s5+s4)<<endl;
    s4+=s5;
    cout<<"s4="<<s4<<endl;
}
```

C:\WINDOWS\system32\cmd.exe

```
s2=old string
s1=old string
s3=old
s4=55555
s3>s4 - 1
s3<s4 - 0
s5=new string
s6=new string
s5==s6 - 1
s5+s4 - new string55555
s4=55555new string
Для продолжения нажмите любую клавишу . . .
```

ФУНКЦИИ

1. Присваивание **assign()**:

- **assign(const string& str)** – присваивает строку str вызывающей строке
- **assign(const string& str, size_type pos, size_type n)**
– присваивает вызывающей строке n символов строки str, начиная с номера pos
- **assign(char* s, size_type n)** – присваивает вызывающей строке n символов строки s в стиле C.

Пример:

s1.assign(s2) равносильно s1=s2

2. Добавление **append()**

- **append(const string& str)** – добавляет строку str к вызывающей строке
- **append (const string& str, size_type pos, size_type n)** – добавляет к вызывающей строке n символов строки str, начиная с номера pos
- **append (char* s, size_type n)** – добавляет к вызывающей строке n символов строки s в стиле C.

Примеры:

```
string s1("STRING");
```

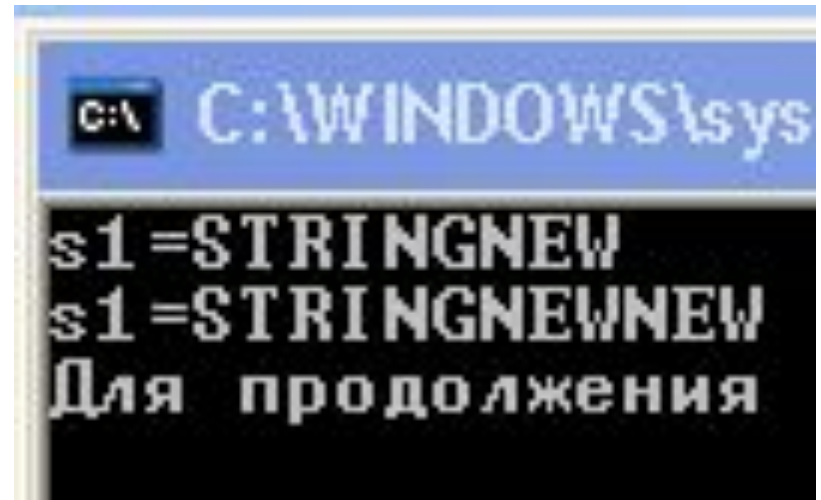
```
string s2("NEW");
```

```
s1.append(s2);// "STRINGNEW"
```

```
s1.append(s2,0,3);// "STRINGNEWNEW"
```

```
#include <iostream>
#include <locale.h>
#include <string>
using namespace std;

void main()
{
    string s1("STRING");
    string s2("NEW");
    s1.append(s2);
    cout<<"s1="<<s1<<endl;
    s1.append(s2,0,3);
    cout<<"s1="<<s1<<endl;
}
```



3. Вставка **insert()**

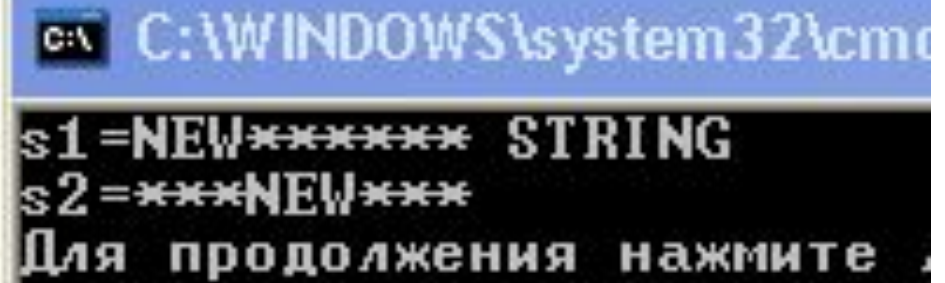
- **insert(size_type pos1, const string& str);** - вставляет строку str в вызывающую строку, начиная с позиции pos вызывающей строки.
- **insert(size_type pos1, const string& str, size_type pos2, size_type n);** - вставляет n символов строки str, начиная с позиции pos2, в вызывающую строку, начиная с позиции pos1 вызывающей строки
- **insert(size_type pos1, const char* s, size_type n);** - вставляет строку в стиле C s в вызывающую строку, начиная с позиции pos вызывающей строки.

Примеры:

```
string s1("NEW STRING"), s2("*****");  
s1.insert(3,s2);  
s2.insert(3,s1,0,3);
```

```
#include <iostream>
#include <locale.h>
#include <string>
using namespace std;
void main()
{
    string s1("NEW STRING"),
    s2("*****");
    s1.insert(3,s2);
    s2.insert(3,s1,0,3);

    cout<<"s1="<<s1<<endl;
    cout<<"s2="<<s2<<endl;
}
```



```
C:\WINDOWS\system32\cmd
s1=NEW***** STRING
s2=***NEW***
Для продолжения нажмите
```

4. Удаление `erase()`

- **`erase(size_type pos=0, size_type n)`** – удаляет `n` символов строки, начиная с `pos`, если `n` не указано, то удаляет строк до конца.

Пример:

```
s1.erase(0,3);
```

5. Очистка всей строки **`clear()`**

```
s1.clear()
```



```
#include <iostream>
#include <locale.h>
#include <string>
using namespace std;
void main()
{
    string s1("NEW STRING"),
    s2("*****");
    s1.insert(3,s2);
    s2.insert(3,s1,0,3);
    cout<<"s1="<<s1<<endl;
    cout<<"s2="<<s2<<endl;
    s1.erase(0,3);
    cout<<"s1="<<s1<<endl;
    s1.clear();
    cout<<"s1="<<s1<<endl;
}
```

C:\ C:\WINDOWS\system32\cmd.exe

s1=NEW***** STRING

s2=***NEW***

s1=***** STRING

s1=

Для продолжения нажмите любую клавишу . . .

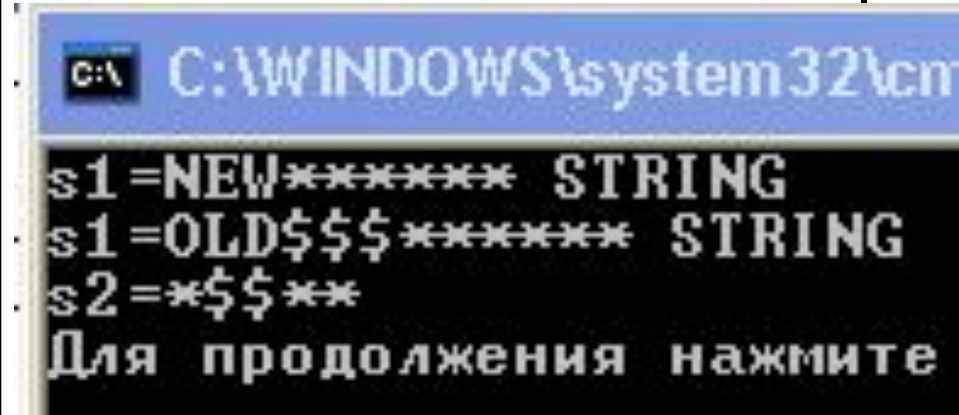
6. Замена части строки **replace()**

- **replace(size_type pos1, size_type n1, const string& str);** - заменяет в вызывающей строке n1 символ, начиная с позиции pos1 на строку str
- **replace(size_type pos1, size_type n1, const string& str , size_type pos2, size_type n2);** - заменяет в вызывающей строке n1 символ, начиная с позиции pos1 на строку n2 символов строки str, начиная с позиции pos2
- **replace(size_type pos1, size_type n1, const char*s, size_type n2);** - заменяет в вызывающей строке n1 символ, начиная с позиции pos1 на n2 символов строки в стиле C s

```
string s3("OLD");
```

```
s1.replace(0,3,s3);
```

```
#include <iostream>
#include <locale.h>
#include <string>
using namespace std;
void main()
{
    string s1("NEW STRING"),
    s2("*****");
    s1.insert(3,s2);
    cout<<"s1="<<s1<<endl;
    string s3("OLD$$$");
    s1.replace(0,3,s3);
    cout<<"s1="<<s1<<endl;
    s2.replace(1,3,s3,4,2);
    cout<<"s2="<<s2<<endl;
}
```



```
C:\WINDOWS\system32\cmd
s1=NEW***** STRING
s1=OLD$$$***** STRING
s2=*$*$*
Для продолжения нажмите
```

7. Обмен содержимого двух строк **swap()**

swap(string &s);

8. Выделение части строки **substr()**

- **string substr (size_type pos=0, size_type n);** - возвращает подстроку вызываемой строки, начиная с символа pos, длиной n

9. Преобразование в строку C **c_str()**

- **const char* c_str() const**

10. Копирование части строки **copy()**

- **size_type copy(char *c, size_type n, size_type pos=0)** – копирует n элементов вызывающей строки в массив c, начиная с позиции pos, ноль-терминатор в массив не заносится, возвращает количество скопированных элементов.

```
#include <iostream>
#include <locale.h>
#include <string>
using namespace std;
void main()
{
    string s1("NEW STRING"),
    s2("*****");

    cout<<"s1="<<s1<<endl;
    cout<<"s2="<<s2<<endl;
    s2.swap(s1);
    cout<<"s1="<<s1<<endl;
    cout<<"s2="<<s2<<endl;
}
```



```
C:\WINDOWS\system32\cmd
s1=NEW STRING
s2=*****
s1=*****
s2=NEW STRING
Для продолжения нажмите
```

11. Поиск подстрок

- **size_type find(const string& str, size_type pos=0) const** – ищет самое левое вхождение строки str в вызывающую строку, начиная с позиции pos, возвращает позицию строки, если она найдена и npos, если строка не найдена (npos – самое большое положительное целое число).
- **size_type find(char c, size_type pos=0) const** – ищет самое левое вхождение символа c в вызывающую строку, начиная с позиции pos, возвращает позицию в строке, если он найден и npos, если символ не найден.

- 12. Сравнение частей строк `compare()`
 - **`int compare(const string&str)const;`** - сравнивает две строк целиком и возвращает значение меньше 0, если вызывающая строка меньше `str`, 0, если они равны и большее 0, если вызывающая строка больше `str`.
 - **`int compare(size_type pos1, size_type n1, const string& str , size_type pos2, size_type n2)const;`** аналогично предыдущему случаю, но сравнивает подстроки в вызывающей строке и строке `str`.
 - **`int compare(size_type pos1, size_type n1, const string& str)const;`** - аналогично предыдущему случаю, но сравнивает подстроку в вызывающей строке и строку `str`.

13. Получение количества элементов в строке

- **size_type size() const**
- **size_type length() const**
- **bool empty () const** – возвращает true, если строка пустая и false в противном случае

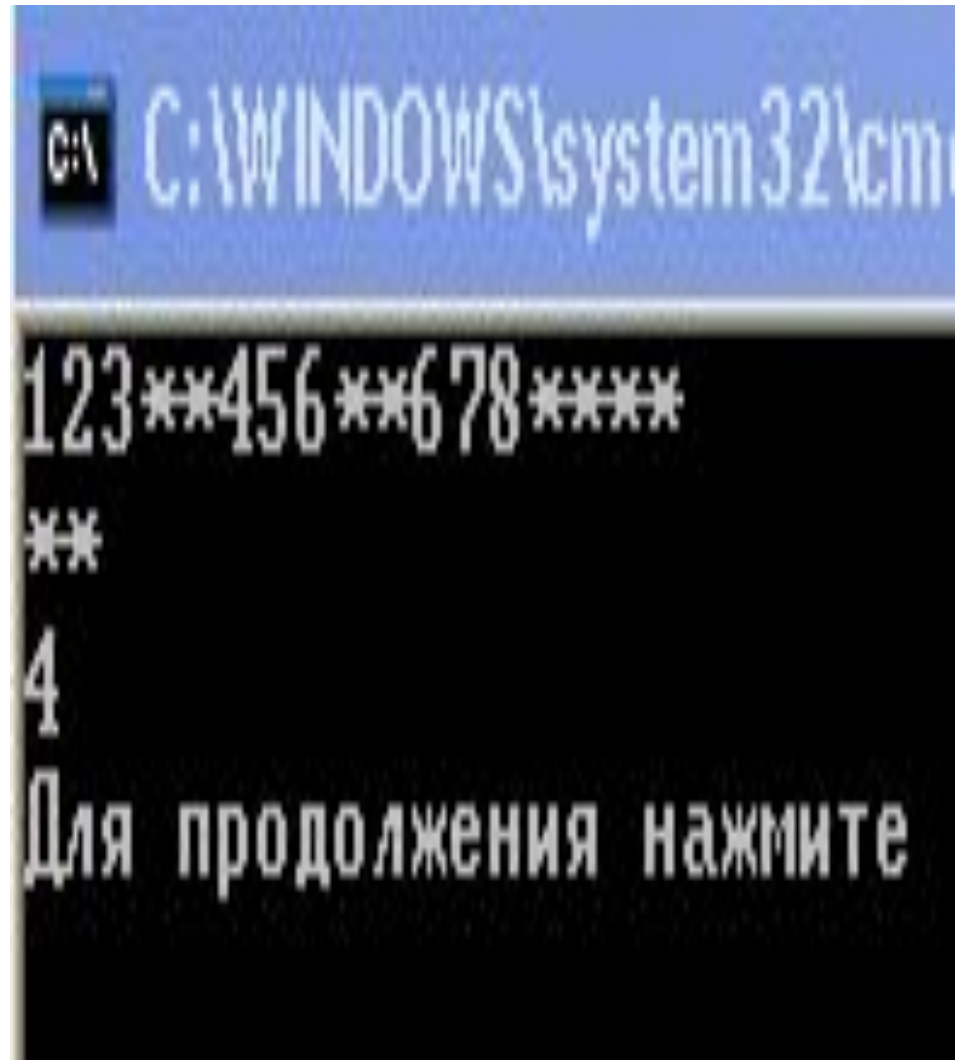
14. Получение количества памяти, занимаемое строкой

size_type capacity() const

Задача. Найти количество вхождений подстроки S1 в строку S

```
#include <iostream>
#include <string>
using namespace std;
void main()
{
    string s,s1;
    int k,i;
    cin>>s;
    cin>>s1;
    k=0;
    i=0;
    while (s.find(s1,k)!=-1)
    {
        k=s.find(s1,k)+s1.length();
        i++;
    }
    cout <<i ;
}
```

```
#include <iostream>
#include <string>
using namespace std;
void main()
{
    string s,s1;
    int k,i;
    cin>>s;
    cin>>s1;
    k=0;
    i=0;
    while (s.find(s1,k)!=-1)
    {
        k=s.find(s1,k)+s1.length();
        i++;
    }
    cout <<i<<endl ;
}
```



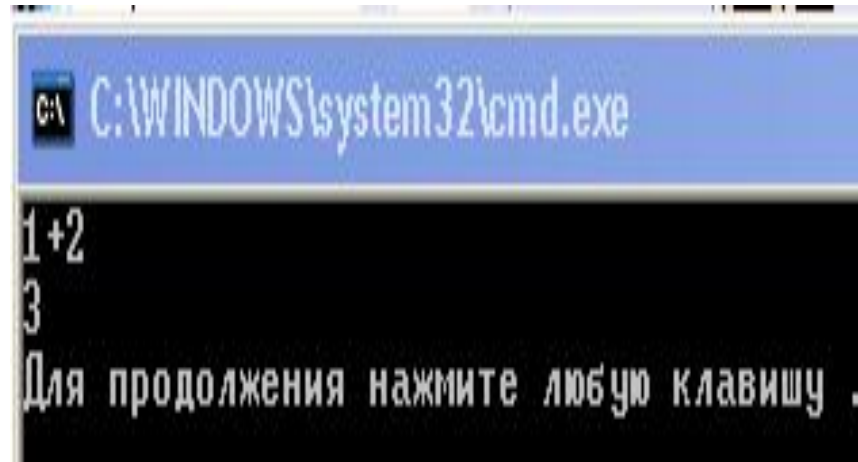
Задача. Дана строка следующего вида $k@m$, где k и m - цифры от 0 до 9, а @-знак операции (+, -, *, /). Вычислить значение данного выражения, если известно, что код символа '0' равен 48.

```

#include <iostream>
#include <string>
using namespace std;

void main()
{
    string s,s1;
    int k1,k2,r;
    cin>>s;
    k1=int(s[0])-48;
    k2=int(s[2])-48;
    switch (s[1])
    {
        case '+':r=k1+k2;break;
        case '-':r=k1-k2;break;
        case '*':r=k1*k2;break;
        case '/':r=k1/k2;break;
    }
    cout <<r<<endl;
}

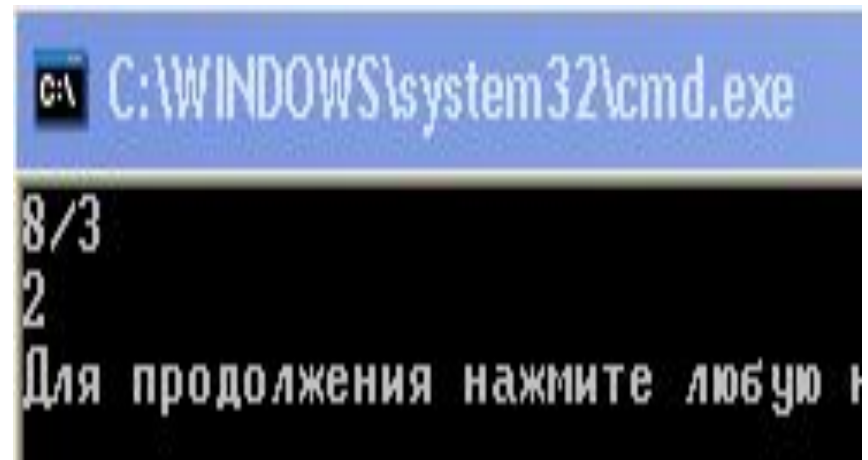
```



```

C:\WINDOWS\system32\cmd.exe
1+2
3
Для продолжения нажмите любую клавишу

```



```

C:\WINDOWS\system32\cmd.exe
8/3
2
Для продолжения нажмите любую клавишу

```

Задача

Дана строка символов. Подсчитать, сколько различных символов встречается в ней. Вывести их на экран.

Введите строку: **11223344**

Различных символов: 4

1 2 3 4

```
#include <iostream>
#include <locale.h>
#include <string>
using namespace std;

void main()
{ int i;
  setlocale(LC_ALL,"rus");
  string s, diff;
  cin>>s; // getline(cin, s);
  diff.clear();
  for(i=0;i<s.size();i++)
    if (diff.find(s[i])==-1) // Если символ в строке не найден, то
      возвращает -1, иначе - позицию
      diff+=s[i];
  cout << "Различных символов " << diff.size()<<endl;
  for (i=0;i<diff.size();i++)
    cout<<diff[i]<<" ";
  cout<<endl;
}
```



C:\WINDOWS\system32\cmd.exe

1234237890*;1

Различных символов 10

1 2 3 4 7 8 9 0 * ;

Для продолжения нажмите любую клавишу . . .

2. Из заданной символьной строки выбрать те символы, которые встречаются в ней только один раз, в том порядке, в котором они встречаются в тексте.


```
#include <iostream>
#include <locale.h>
#include <string>
using namespace std;

void main()
{ int i,j;
  setlocale(LC_ALL,"rus");
  string s;
  bool f;
  getline(cin, s);
  int length=s.size();
```

```
for(i=0;i<length;i++)
{
  // считаем, что очередной
  СИМВОЛ встречается один
  раз

  f=true;
  for (j=0; j<length &&
f;j++)
      f=!(s[i]==s[j] && i!=j);
  if (f) cout<< s[i]<<" ";
}
}
```



C:\WINDOWS\system32\cmd.exe

112342053

4 0 5 Для продолжения нажмите любую клавишу

3. Дана строка S , которая содержит одно слово. Проверить, будет ли оно читаться одинаково справа налево и слева направо (т.е. является ли оно палиндромом).

```
#include <iostream>
#include <locale.h>
#include <string>
using namespace std;

void main()
{ int i,j;
  setlocale(LC_ALL,"rus");
  string s; bool f;
  cin>>s;
  int length=s.size();
  int length2=length/2;
  i=0;
```

```
while (i<length2 &&
s[i]==s[length-i-1])
    i++;
    if (i>=length2)
        cout<<"
Палиндром"<<"\n";
    else
        cout<<"Нет"<<"\n";
}
```

C:\ C:\WINDOWS\system32\cmd.exe

12345

Нет

Для продолжения нажмите любую клавишу . . .

C:\ C:\WINDOWS\system32\cmd.exe

123321

Палиндром

Для продолжения нажмите любую клавишу . . .

4. Дана строка S. Найти количество букв в самом длинном слове в данной строке. Знак препинания приравнять к букве и считать допустимой частью слова.

```

void main()
{
    setlocale(LC_ALL,"rus");
    string s; int max_len, cur_len;
    char sl[20];
    int poz,len;
    getline(cin, s);
    s+=" ";          // искусственный прием для выделения последнего слова
    max_len=0;
    while (!s.empty())
    {
        while (!s.empty() && s[0]==' ') //Удаление пробелов в начале строки
            s.erase(0,1);
        if (!s.empty())
        {   poz=s.find(" ");          // последний символ ближайшего слова
            len=s.copy(sl,poz,0);      // sl - слово
            sl[len]='\0';
            cur_len=strlen(sl);
            if (cur_len>max_len)
                max_len=cur_len;
            s.erase(0,poz);
        }
    }
    cout<<max_len<<endl;
}

```

C:\ C:\WINDOWS\system32\cmd.exe

Желаю счастья, здоровья, успехов.

9

Для продолжения нажмите любую клавишу . . .

Использование датчика случайных чисел для заполнения массивов

```
#include "stdio.h"
#include "stdlib.h"
void main()
{int a[100];
int i;
for(i=0;i<100;i++)
    a[i]=rand() % 200 -100;
for(i=0;i<100;i++)
{ if (i % 10==0) printf("\n");
printf("%4d",a[i]);
}
printf("\n");
}
```

C:\WINDOWS\system32\cmd.exe

```
-59 -33 34 0 69 24 -22 58 62 -36
5 45 -19 -73 61 -9 95 42 -73 -64
91 -96 2 53 -8 82 -79 16 18 -5
-53 26 71 38 -31 12 -33 -1 -65 -6
3 -89 22 33 -27 -36 41 11 -47 -32
47 -56 -38 57 -63 -41 23 41 29 78
16 -65 90 -58 -12 6 -60 42 -36 -52
-54 -95 -10 29 70 50 -94 1 93 48
-71 -77 -16 54 56 -60 66 76 31 8
44 -61 -74 23 37 38 18 -18 29 41
```

Для продолжения нажмите любую клавишу . . .

C:\WINDOWS\system32\cmd.exe

```
-59 -33 34 0 69 24 -22 58 62 -36
5 45 -19 -73 61 -9 95 42 -73 -64
91 -96 2 53 -8 82 -79 16 18 -5
-53 26 71 38 -31 12 -33 -1 -65 -6
3 -89 22 33 -27 -36 41 11 -47 -32
47 -56 -38 57 -63 -41 23 41 29 78
16 -65 90 -58 -12 6 -60 42 -36 -52
-54 -95 -10 29 70 50 -94 1 93 48
-71 -77 -16 54 56 -60 66 76 31 8
44 -61 -74 23 37 38 18 -18 29 41
```

Для продолжения нажмите любую клавишу . . .

```
#include "stdio.h"
#include "stdlib.h"
#include "time.h"
void main()
{int a[100];
int i;
time_t t;
srand((unsigned) time(&t));
for(i=0;i<100;i++)
    a[i]=rand() % 200 -100;
for(i=0;i<100;i++)
{ if (i % 10==0) printf("\n");
printf("%4d",a[i]);
}
printf("\n");
}
```

```
C:\ C:\WINDOWS\system32\cmd.exe

-2 -3 -41 -30 49 -9 -2 75 19 -32
8 24 74 84 -32 -23 24 -91 75 -4
-89 -68 -89 -30 -72 29 93 -15 77 -74
-21 -21 51 65 68 39 28 7 28 -65
51 40 15 -54 9 -63 -47 58 48 85
-12 40 6 -97 37 -25 13 -50 -65 -42
-87 -22 61 -46 9 -10 98 -100 9 -49
-37 -50 79 68 -81 -60 61 79 -98 1
-26 27 -96 23 -20 5 -33 -77 -94 -48
-88 88 22 44 -90 -93 -71 -97 -17 -33
Для продолжения нажмите любую клавишу . . .
```

```
C:\ C:\WINDOWS\system32\cmd.exe

-97 66 -11 -68 35 -26 93 -93 -98 38
75 34 22 79 21 -60 31 92 -78 -78
-41 2 34 61 71 30 38 -49 -29 88
-95 27 92 76 7 26 76 54 84 88
-88 -2 -95 30 5 -20 -6 -79 68 30
-84 58 -88 -40 77 -72 -85 -53 68 19
27 -15 83 58 90 -72 -79 79 -57 -83
-59 1 55 -29 -48 -5 -64 28 83 -37
63 -10 -31 -82 80 2 30 -78 70 -7
-80 76 -84 -32 4 -6 27 95 -68 12
Для продолжения нажмите любую клавишу . . .
```