

Работа с текстовыми файлами

C++

Файл

- **Файл** – поименованная область на внешнем носителе

Файл можно представить как конечное количество последовательных байтов

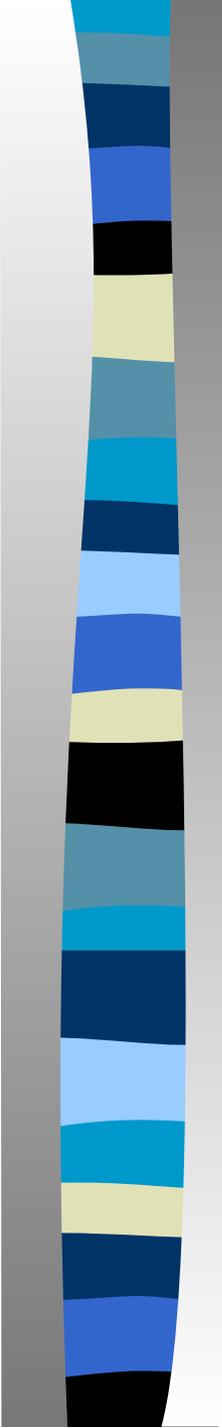
- **Устройства**, такие как дисплей, клавиатура, принтер, можно рассматривать как частные случаи файлов



Поток (stream)

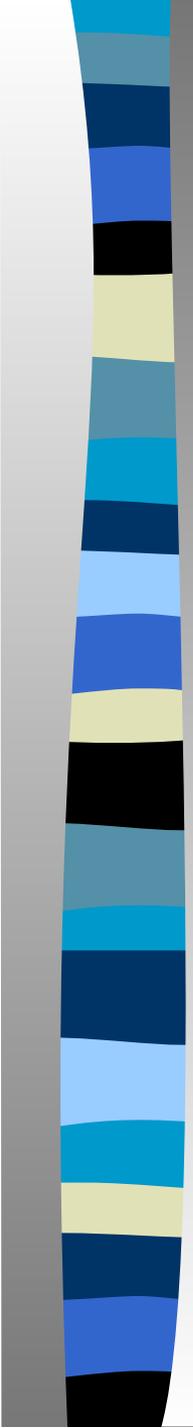
- **Поток** – это абстрактное понятие, относящееся к любому переносу данных от источника к приёмнику
- **Поток** определяется как последовательность байтов и не зависит от конкретного устройства, с которым производится обмен





Буфер

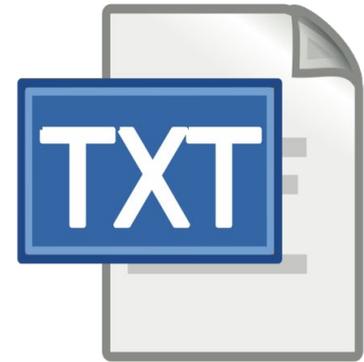
- Для повышения скорости передачи данных обмен с потоком производится через специальную область оперативной памяти – **буфер**
- Буфер выделяется для каждого открытого файла
- При записи в файл данные направляются в буфер и там накапливаются, а затем происходит их передача на внешнее устройство
- При чтении из файла данные вначале считываются в буфер (сколько помещается)



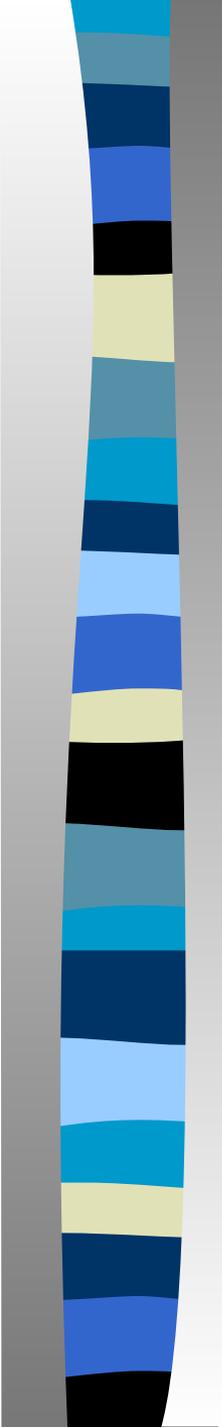
Файлы по принципам работы делятся на:

- **Текстовые** (возможен только последовательный доступ)
Содержимое таких файлов пользователь может прочитать и отредактировать в текстовом редакторе
- **Двоичные** (возможен как последовательный, так и прямой доступ)
Интерпретация таких файлов возлагается на специализированные программные средства

Текстовые файлы:



- Позволяют обрабатывать информацию только последовательно, от начала до конца
- Новую информацию можно добавлять только в конец файла



Двоичные файлы:

- Для них можно использовать как последовательное чтение или запись, так и режим прямого доступа к любой компоненте файла с заданным номером
- В режиме прямого доступа можно прочесть одну из компонент с любым номером, или изменить (перезаписать) компоненту с любым номером

Текстовые файлы

- Имеют расширение **txt**
- Создать файл и записать в него информацию можно заранее в текстовом редакторе
- Текстовый файл может содержать как текстовую, так и числовую информацию
- Относятся к файлам **последовательного** доступа
- Обработываются с преобразованием посимвольно или построчно, а так же могут быть прочитаны сразу полностью

Открытие текстового файла для чтения:

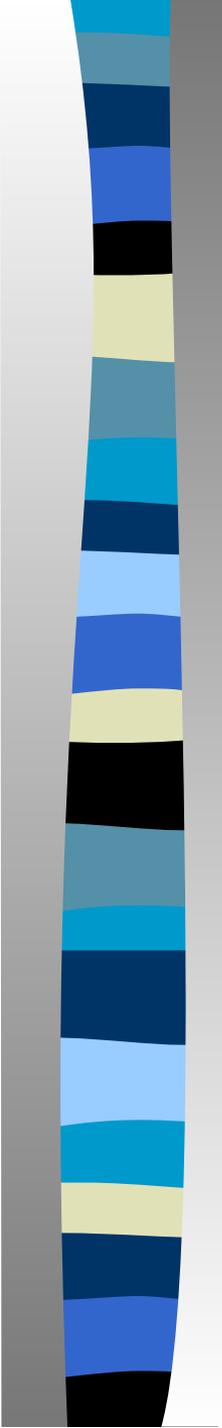
```
ifstream F1;  
F1.open("1.txt", ios::in | ios::nocreate );  
if (!F1) cout<<"Ошибка открытия файла!";  
    // проверим, существует ли файл
```

После удачного открытия файла (в любом режиме) в переменной F будет храниться **true**, в противном случае **false**. Это позволит проверить корректность операции открытия файла.

Открытие текстового файла для записи:

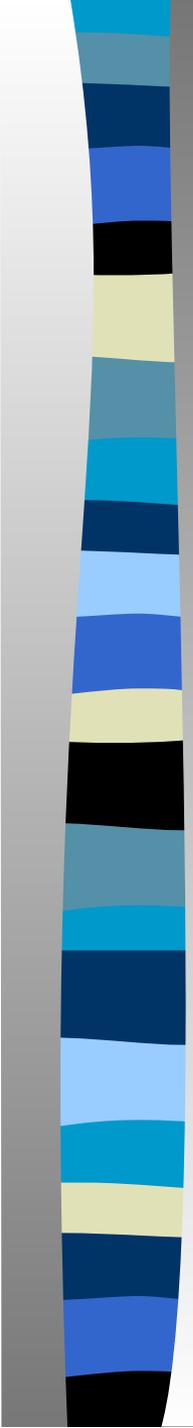
```
ofstream F2;  
F2.open("2.txt", ios::out );
```

Параметр mode может отсутствовать, в этом случае файл открывается в режиме по умолчанию для данного потока



Файл может открываться в следующих режимах:

- В режиме **чтения**
- В режиме **запись** (перезапись)
- В режиме добавления информации в конец файла



`ios::in` — открыть файл в режиме чтения данных; режим по умолчанию для потоков

`ifstream`;

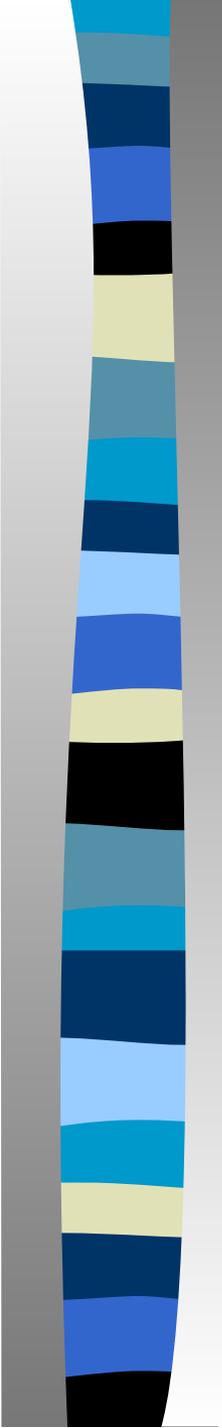
`ios::out` — открыть файл в режиме записи данных (при этом информация о существующем файле уничтожается); режим по умолчанию для потоков `ofstream`;

`ios::app` — открыть файл в режиме добавления данных в конец файла;

`ios::ate` — передвинуться в конец уже открытого файла;

`ios::trunc` — очистить файл, то же происходит в режиме `ios::out`;

`ios::nocreate` — не выполнять операцию открытия файла, если он не существует.



Заккрытие файла

```
F1.close();
```

- Эта функция удаляет связь файловой переменной с файлом на диске и аналогична сохранению файла
- Во избежание ошибок при работе с файлами, для каждой процедуры открытия файла в программе должна присутствовать команда close

Запись данных в файл

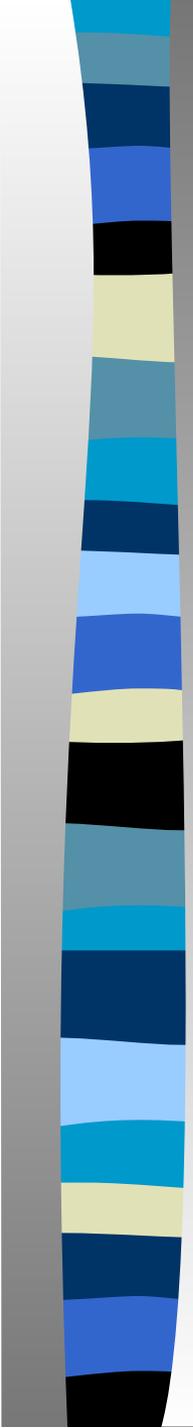
- После открытия файла в режиме записи будет создан пустой файл, в который можно будет записывать информацию

```
ofstream F2;  
F2.open("2.txt", ios::out );  
char str[80];  
cout<<"Введите строку";  
cin>>str;  
F2<<str;  
F2.close();
```

Чтение данных из файла

- После открытия файла в режиме чтения можно прочитать из него информацию

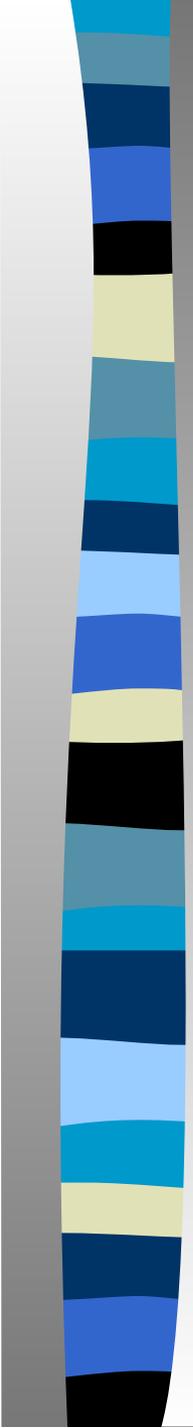
```
ifstream F1;
F1.open("1.txt", ios::in | ios::nocreate );
if (!F1)
    { cout<<"Ошибка открытия файла!";
      return 0;
    }
char str[80];
F1>>str; // прочитать все данные
cout<<str; // вывести на экран
F1.close();
```



Чтение данных построчно

- В текстовом файле обычно находится много строк, поэтому чтение данных происходит в цикле

```
ifstream F1;
F1.open("1.txt", ios::in | ios::nocreate );
if (!F1)
    {   cout<<"Ошибка открытия файла!";
        return 0;
    }
char str[80];
while (!F1.eof()){
    F1.getline(str, 80);
    cout<<str<<endl;    }
F1.close();
```



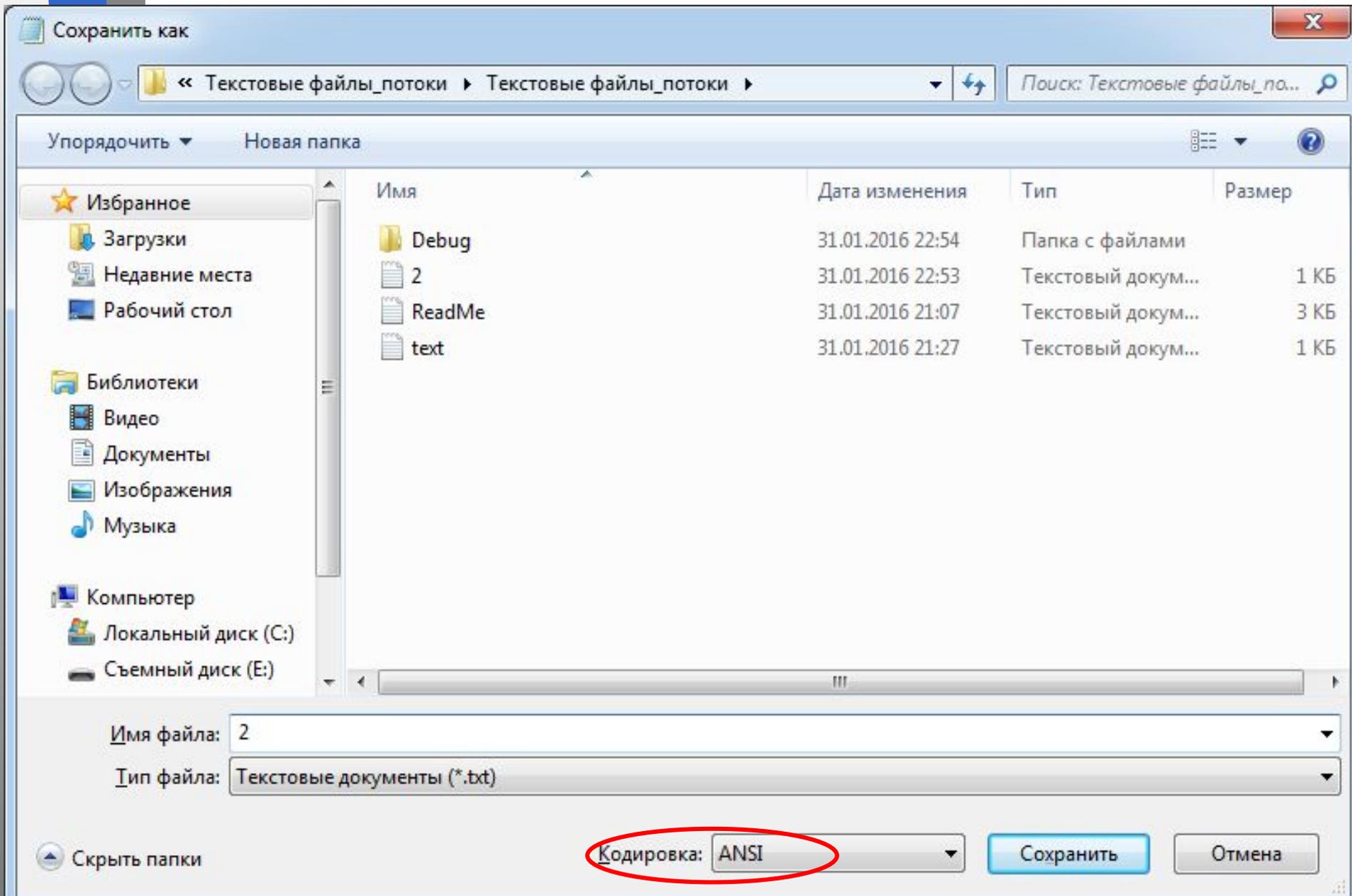
Работа с текстовым файлом

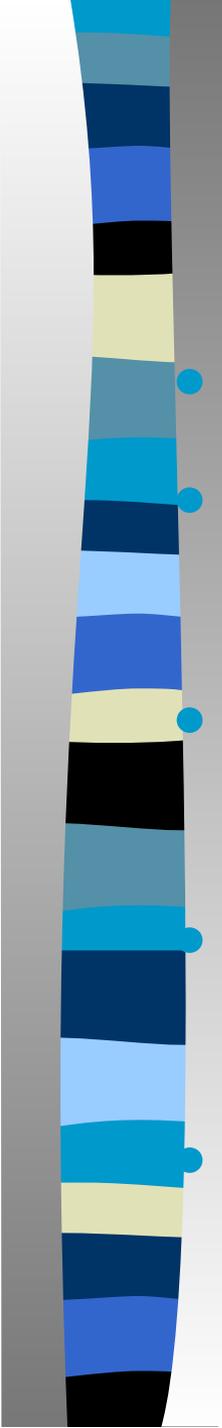
1. Создание файла в редакторе
2. Объявления потока для чтения (файловой переменной)
3. Открытие файла в режиме чтения
4. Обработка информации из файла посимвольно или построчно в цикле
5. Вывод на экран всех данных или выборочно
6. Может быть запись измененной информации в конец файла или в другой файл (заранее открыть в режиме записи)
7. Закрытие всех файлов

Задача 1

Дан текстовый файл. Вывести на экран строки из файла, начинающиеся с букву «Т».

Сохранение файла

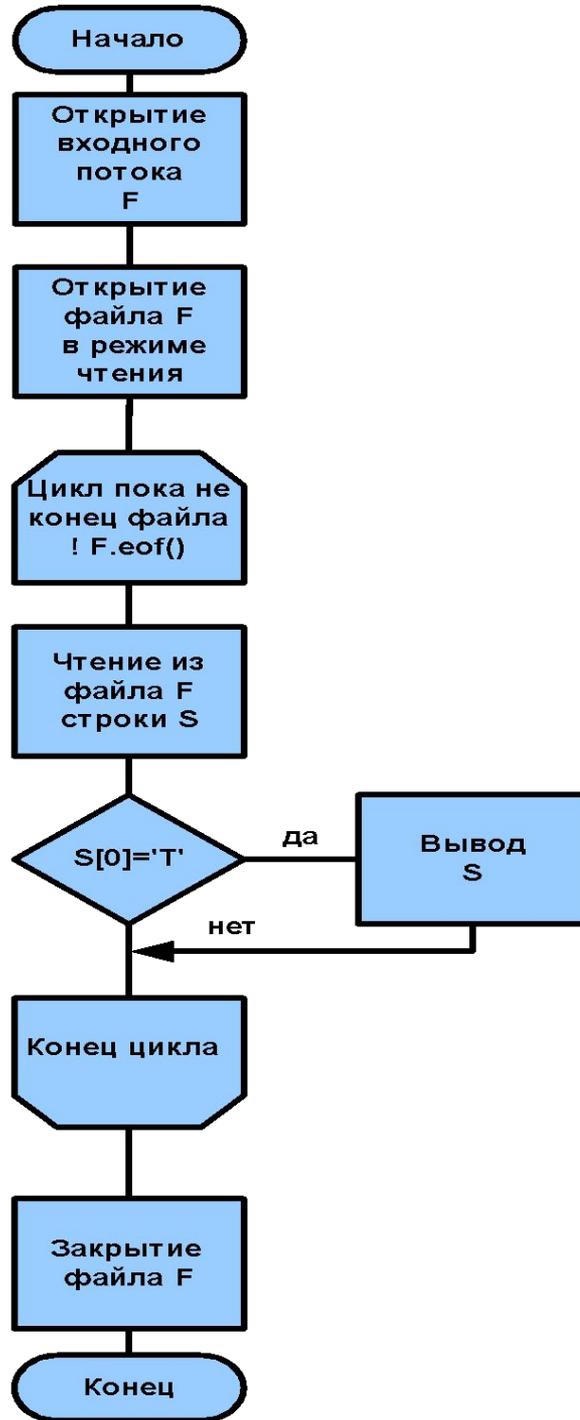




Математическая формулировка:

- Открыть входной поток
- Открыть исходный файл в режиме чтения
- Читать информацию построчно в цикле
- В каждой считанной строке проверять первую букву
- Если первая буква 'Т', то выводить эту строку на экран

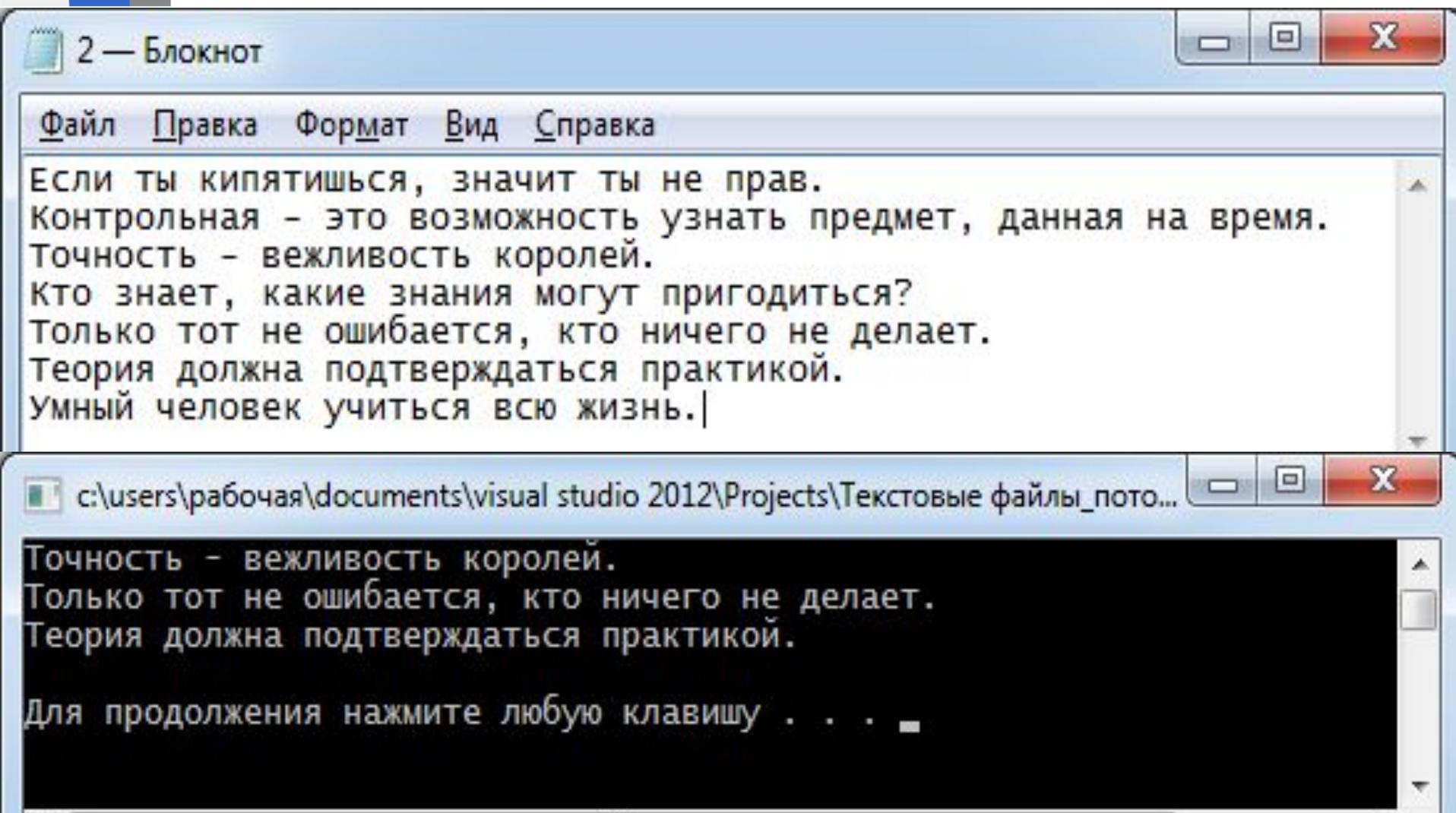
Алгоритм



Текст программы:

```
void main()
{  SetConsoleCP(1251);
   SetConsoleOutputCP(1251);
   char S[80];
   ifstream F1;
   F1.open("2.txt", ios::in);
   while (!F.eof())
   {   F.getline(S, 80);
       if (S[0] == 'T')
           cout<<S<<endl;
   }
   F.close();
   system("pause"); }
```

Результаты работы программы:



Задача 2

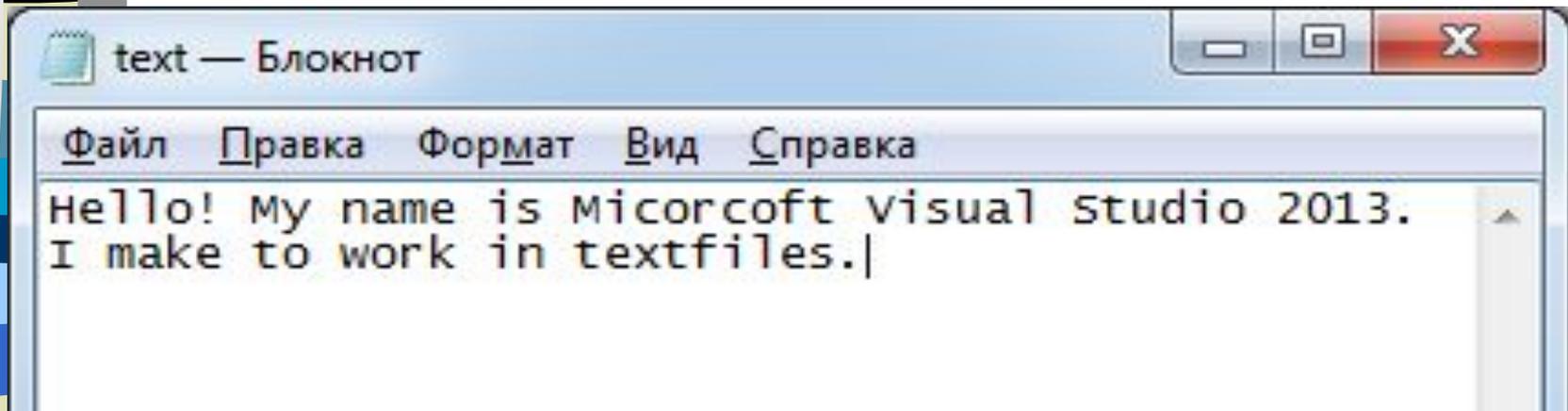
Дан текстовый файл. Определить, встречается ли в файле заданное слово.

Текст программы:

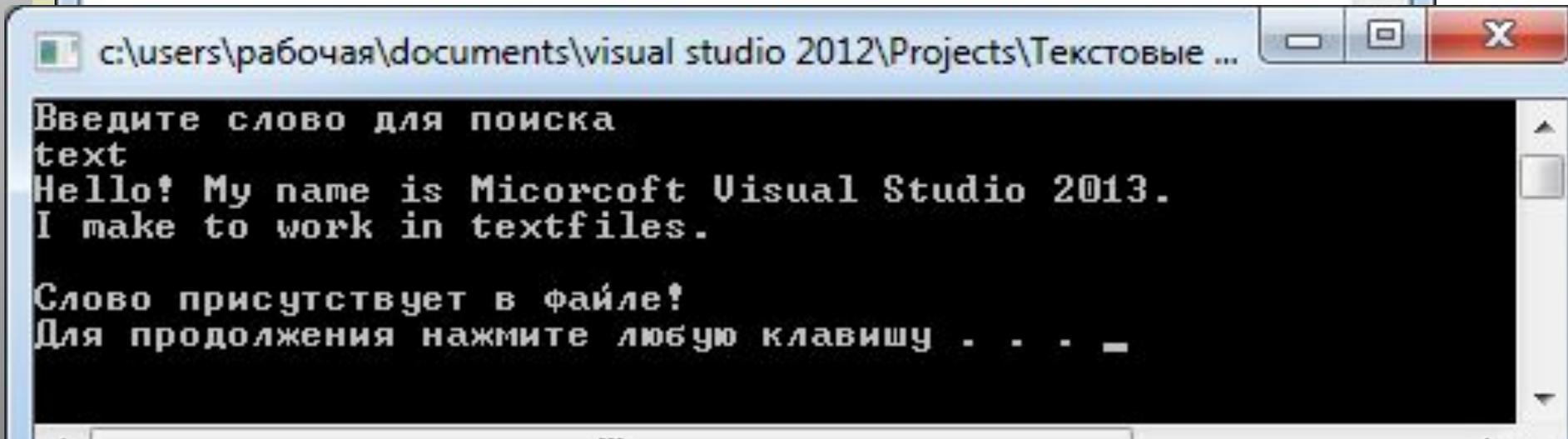
```
void main()  
{  char word[80], str[80];  
   SetConsoleCP(1251);  
   SetConsoleOutputCP(1251);  
   cout<<"Введите слово для поиска";  
   cin>>word;  
   bool fl = false;  
   ifstream F;  
   F.open("text.txt", ios::in);
```

```
while (!F.eof())
{
    F.getline(str, len);
    cout<<str<<endl;
    if (strstr(str,word))
    {
        f1 = true;
    }
}
F.close();
if (f1)
    cout<<"Слово присутствует в файле!"<<endl;
else
    cout<<"Слово отсутствует в файле!"<<endl;
}
```

Результаты работы программы:



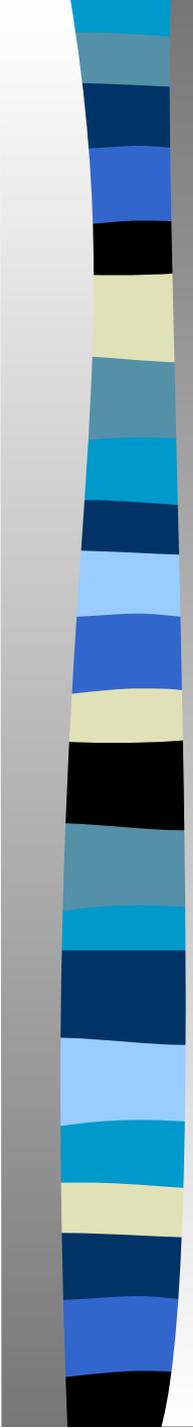
```
text — Блокнот  
Файл  П_равка  Ф_ормат  В_ид  С_правка  
Hello! My name is micorcraft visual studio 2013.  
I make to work in textfiles.|
```



```
c:\users\рабочая\documents\visual studio 2012\Projects\Текстовые ...  
Введите слово для поиска  
text  
Hello! My name is Micorcraft Visual Studio 2013.  
I make to work in textfiles.  
  
Слово присутствует в файле!  
Для продолжения нажмите любую клавишу . . . _
```

Задача 3

Дан текстовый файл. Строки, в которых встречается заданное слово (слово), записать в другой файл.



Математическая формулировка:

- Необходимо открыть входной поток и исходный файл в режиме чтения
- Заранее новый файл открывается для записи (он будет создан в этом режиме)
- Считать данные построчно в цикле
- Проверить каждую строку и определить, встречается ли в ней заданное слово
- Если слово присутствует, то записать эту строку в новый файл

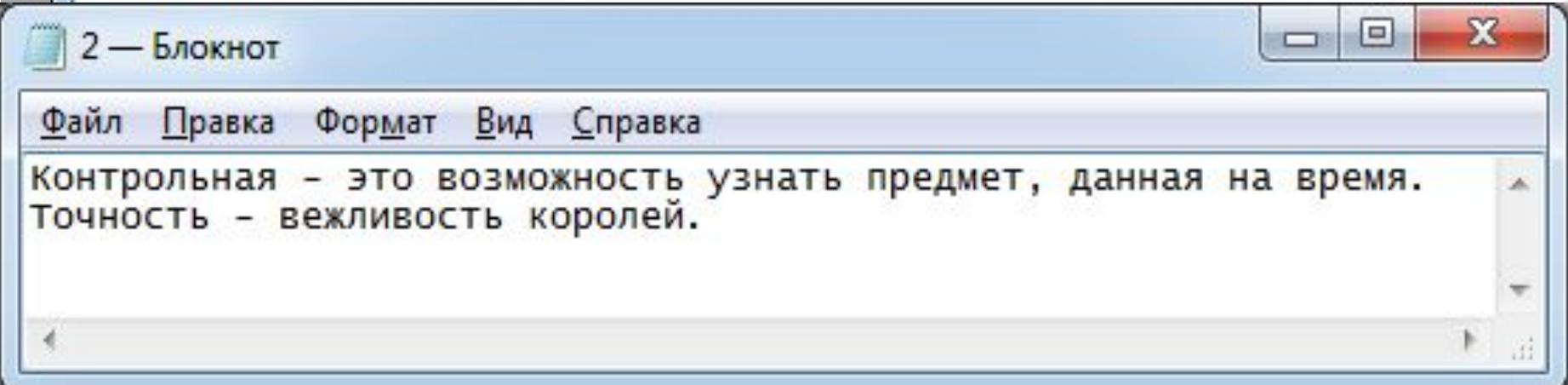
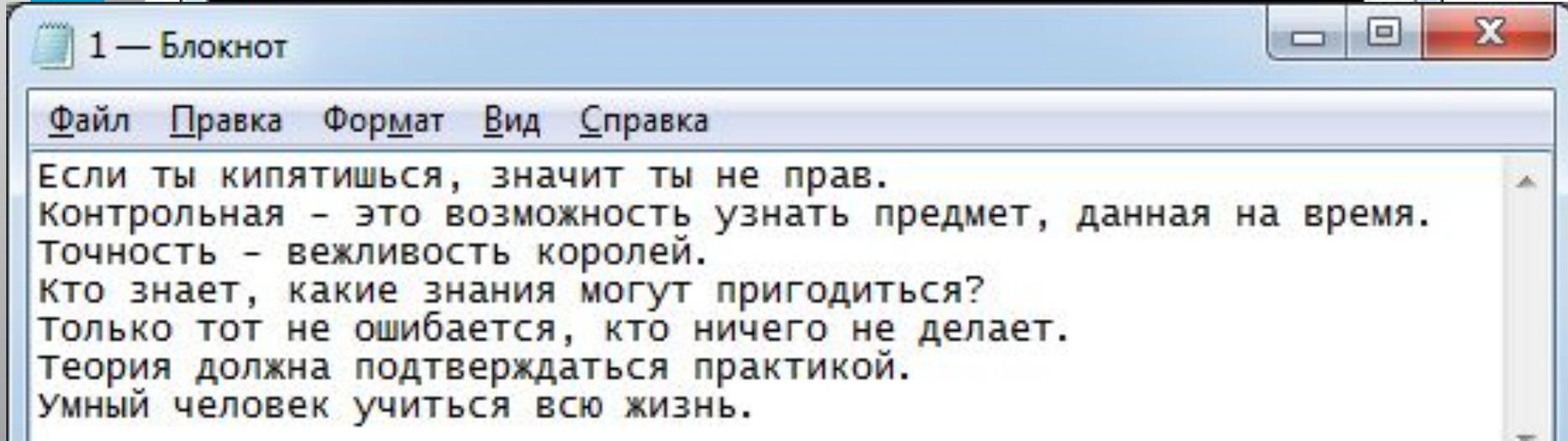
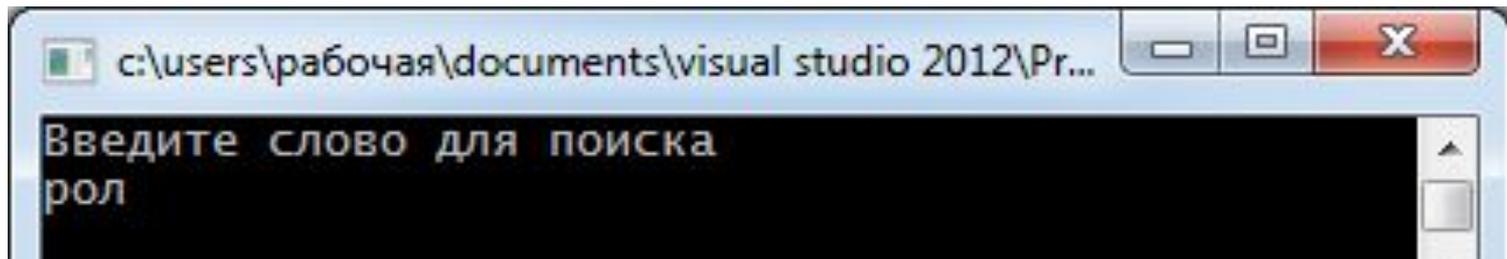
Алгоритм

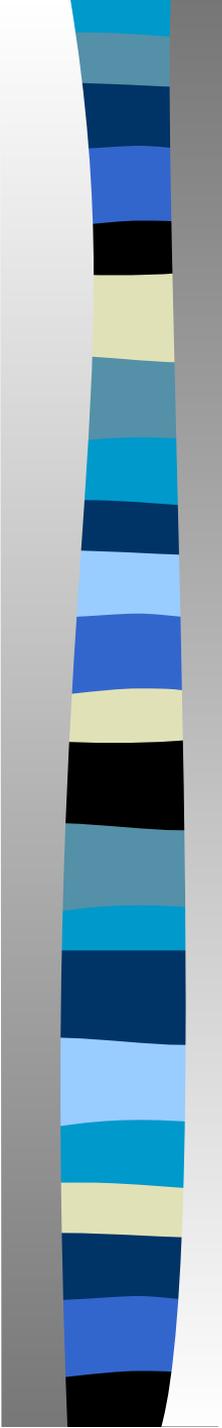


Текст программы:

```
ifstream F1;
F1.open("1.txt", ios::in);
ofstream F2;
F2.open("2.txt", ios::out);
while (!F1.eof())
{
    F1.getline(str, 80);
    if (strstr(str, word))
        F2<<str<<endl;
}
F1.close();
F2.close();
```

Результаты работы программы

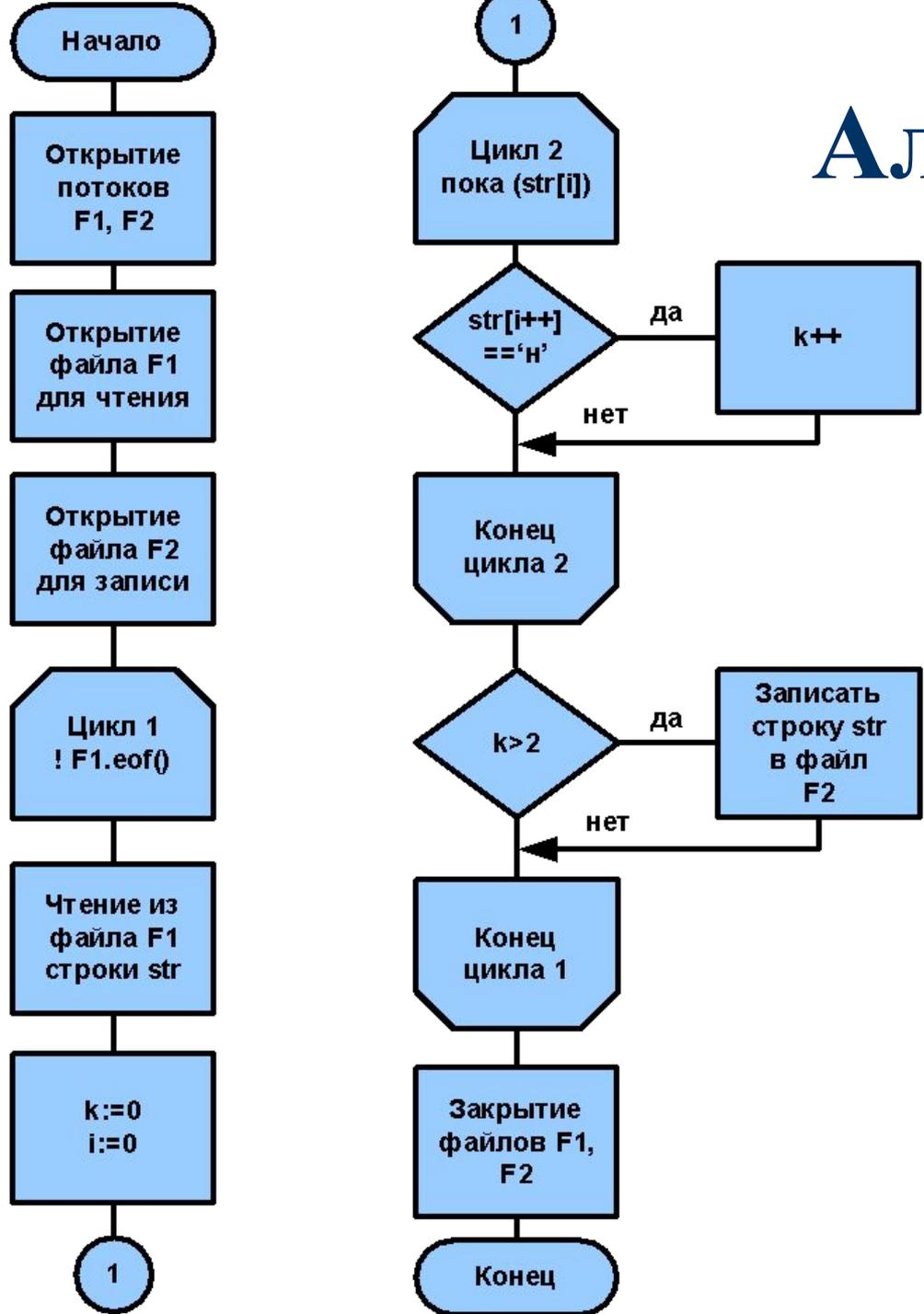




Задача 4

Дан текстовый файл. Строки, в которых встречается больше 2 букв 'н', записать в другой файл.

Алгоритм



Текст программы:

```
ifstream F1;  
F1.open("1.txt", ios::in);  
ofstream F2;  
F2.open("2.txt", ios::out);  
while (!F1.eof())  
{  
    F1.getline(str, 80);  
    i = 0; k = 0;  
    while (str[i])  
    {  
        if (str[i++] == 'H') k++;  
    }  
    if (k>2)  
        F2<<str<<endl;  
}  
F1.close();    F2.close();
```

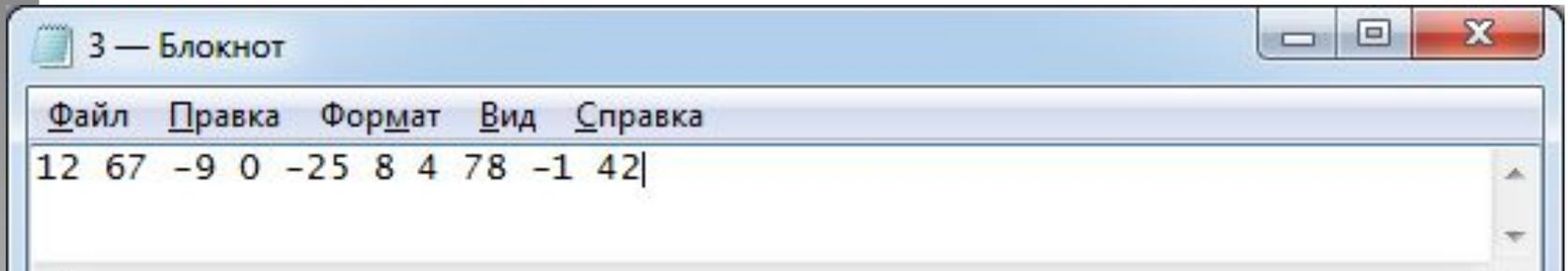
Задача 5

Создать текстовый файл, внести в него 10 целых чисел, каждое через один пробел. Прочитать эти данные в одномерный массив и вывести его на экран. Найти сумму элементов массива.

Текст программы:

```
int A[10];
ifstream F;
F.open("3.txt", ios::in);
int i = 0, sum = 0;
while (i<10)
{
    F>>A[i];
    sum += A[i];
    cout<<A[i]<<'\t';
    i++;
}
F.close();
cout<<"sum = "<<sum<<endl;
```

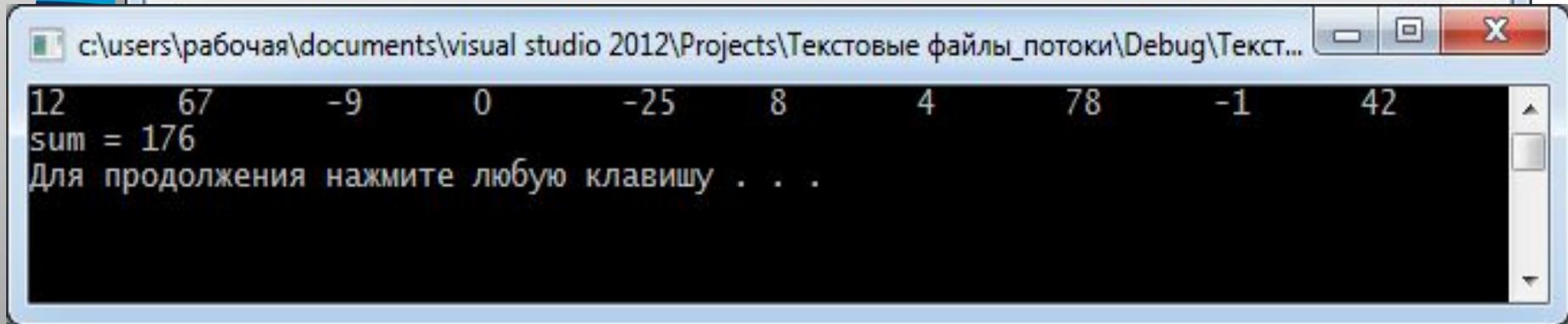
Результаты работы программы



3 — Блокнот

Файл Правка Формат Вид Справка

```
12 67 -9 0 -25 8 4 78 -1 42|
```



c:\users\рабочая\documents\visual studio 2012\Projects\Текстовые файлы_потoki\Debug\Текст...

```
12      67      -9      0      -25      8      4      78      -1      42  
sum = 176  
Для продолжения нажмите любую клавишу . . .
```

Задача

Создать два текстовых файла, внести в них по 5 строк. Разработать программу для получения результирующего файла, в котором будут чередоваться строки из 1-го и 2-го файла.

