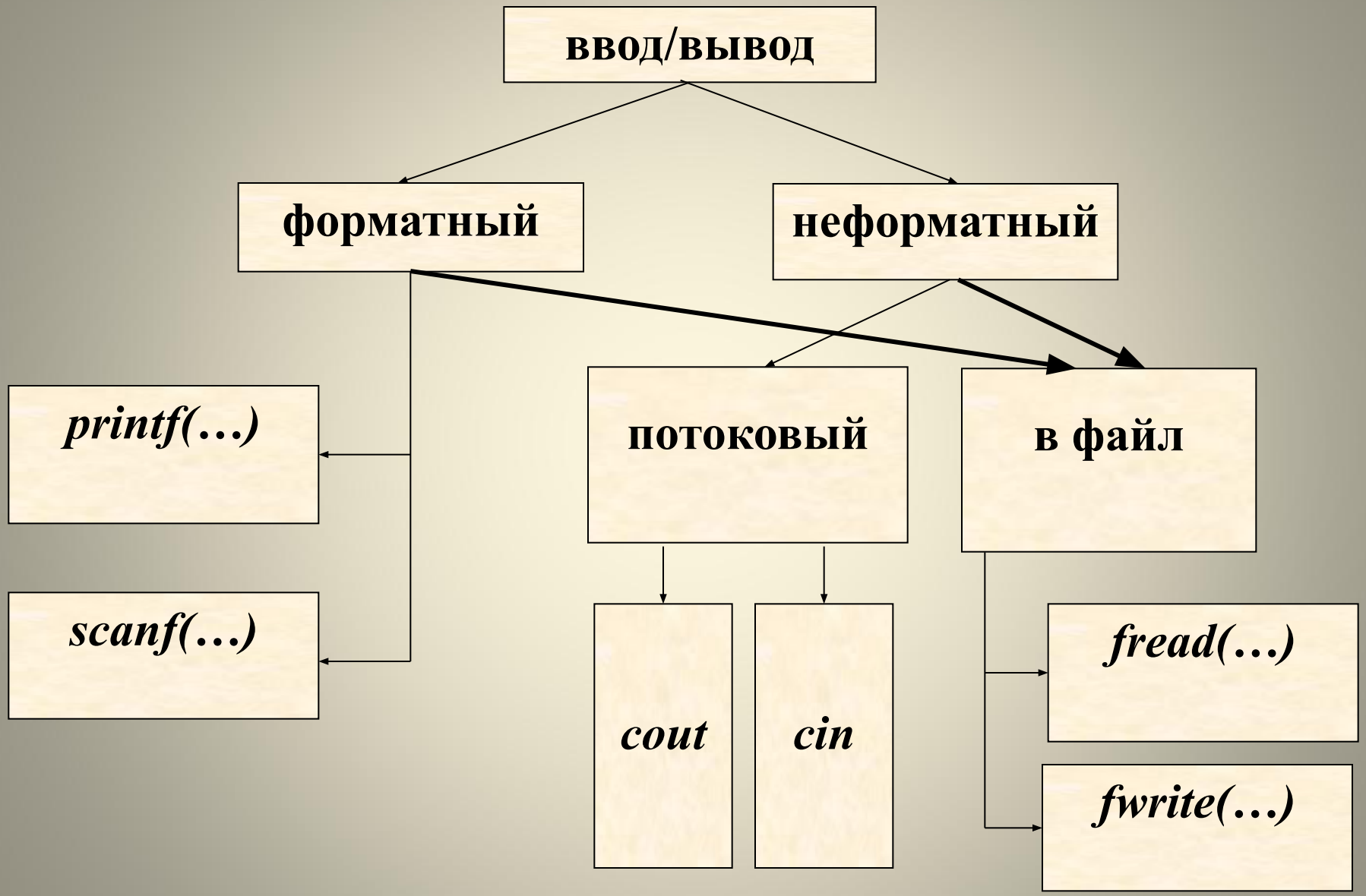
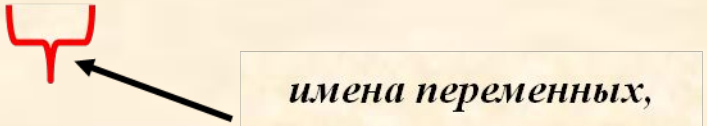
Операторы ввода и вывода



примеры форматного ввода/вывода

```
# include <stdio.h>
…
                                        …
scanf (”%d%f”, &k, &a  );   // с клавиатуры вводятся эти данные: k=5, a=0.527
```


```
printf (”\nk=%d a=%f”, k, a  ); // они же отображаются на экране: k=5, a=0.527
```




```
//пример главной функции некоторой программы
…
//формат вывода и формат представления числа – это не одно и то же
void main (void)
 {
   clrscr();
   int k=29555;     //задано целое десятичное значение к
   printf(“\nдесятичное %d восьмеричное %o шестнадцатиричное %x”,
            k, k, k);   //отпечатано в трех различных форматах
 }//end main
…
```

другие функции форматного ввода/вывода

1)
// см. в справочной системе компилятора:
cprintf(“это специализированная функция вывода***”);***
//буква с показывает, что обрабатываются символы

2)
// см. в справочной системе компилятора:
putchar(“это специализированная функция вывода***”);***
//за один шаг работы отправляет на устройство вывода один символ
getchar( ); **//наоборот забирает один символ из буфера ввода**

//пример функции обработки символов

```
void main (void) {
    clrscr(); int ch;
    while((ch= getchar()) != ‘\n’)
        printf(“%c”, ch);
  }
```

//для Си визуальное представление символов символами
//не отражает их реального внутреннего

Пример программы: чтение данных.

```
# include <stdio.h>
  main ( )//по умолчанию имеет тип int
  {
    int a, b, c e;
    printf (”Введите данные: ”);
    scanf (”%d%i%o%u%x”, &a, &b, &c, &d, &e,);
          // Все целые.
    printf (”%d %d %d %d %d\n”, a, b, c, d, e);
    return 0;
  }
```

Результаты работы операторов печати ниже:

Чтение символов и строк. Пример программы

```
# include <stdio.h>
  int main ( )
  {
    char x, y[7];
    printf (“Введите строку: “);
    scanf (“%c%s”, &x, &y);
    printf (“\nЗдесь символ: %c\n”, x);
    printf (“Здесь строка: %s”, y);
    return 0;
  }
```

Результаты работы программы:

“Введите строку:” ***кафедра***

“здесь символ:” ***к*** **- это выводит программа**

“здесь строка:” ***афедра***

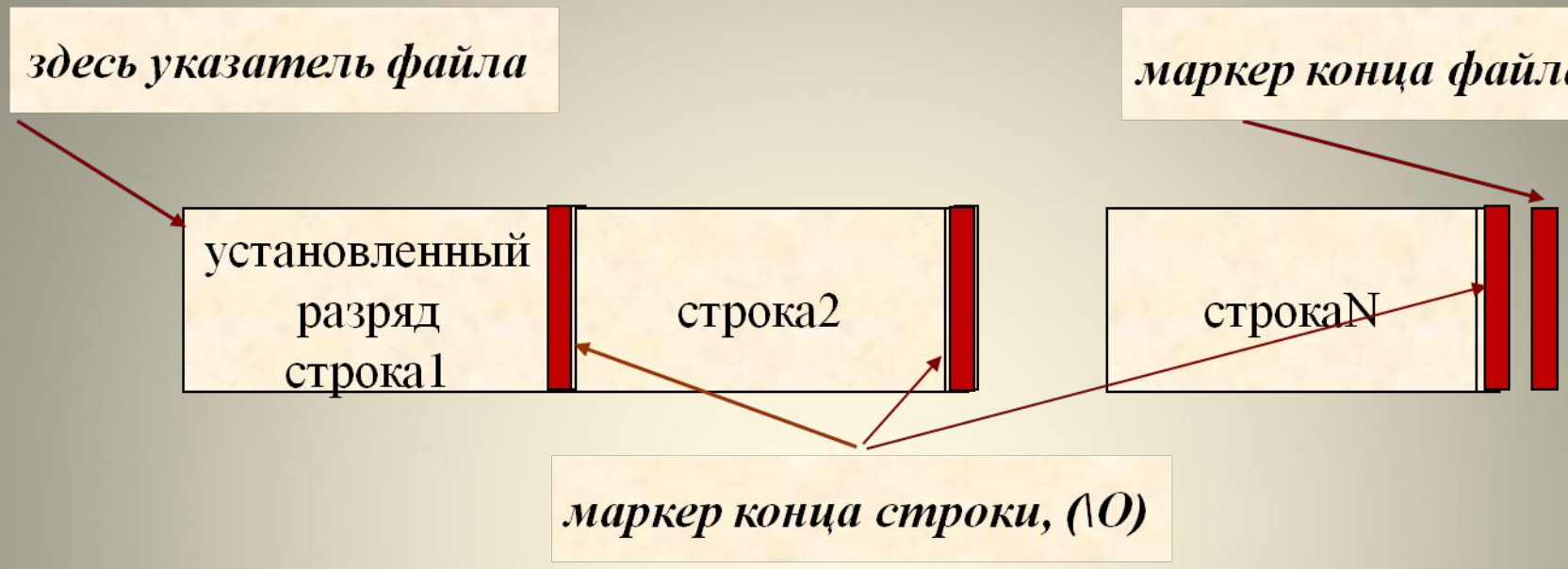
Операторы потокового ввода

```
cout << a << b << c << d
```

```
cin >> a >> b >> c >> d
```

```
#include <conio.h> //пример 1
#include <iostream.h>
 void main ()
   { clrscr();
      int x=75;
      cout << x ++ << " " << x+1 << " ";
      cout << x;
    }// в итоге: 75 76 77. П о ч е м у ?
#include <conio.h> //пример 2
#include <iostream.h>
void main ()
   { clrscr();
      int i=0; x[7];
      while (cin >> x[i++], i!=7);
      for (i=0; i<=6;i++)
      cout << x[i] << " ";
    }// сделаем анализ !
```

ввод/вывод данных в файл



Функция чтения данных из файла:

size_t fread(void *ptr, size_t size, size_t n, FILE *stream);

Функция записи данных из файла:

size_t fwrite(const void *ptr, size_t size, size_t n, FILE *stream);

Программа чтения данных из файла

```
# include <stdio.h> // первая часть
  int main (void)
    {
       int k,                    // Данные будут прочитаны из файла и
           k1,                   // могут использоваться в данной программе.
           k2;
     float x;
     long l;
     char ch,
     str[15];
     file *f_cht;              // Указатель на файл для чтения.
     int col;                  // Возвращаемое значение fscanf.
                               // Открываем файл «f1.dat» для чтения.
     f_cht=fopen ("f1.dat", "r");
       if (f_cht==NULL) // NULL–нулевой указатель (означает ошибку).
           {
             printf ("\nФайл f1.dat для чтения не открыт. Ошибка!");
             return 1;}
// далее продолжение во 2-й части
```

// Чтение данных файла (продолжение 1-й части)

```
col=fscanf (f_cht, %x %d %o %lf %f %c %s %c %s", &k, &k1, &k2, &l,
        &x, &ch, str, &str[3], &str[4]);
        if (col!=9)
      {
       printf ("\nДанные прочитаны с ошибками.");
       return 2;
      }
```

// Закрываем файл.

```
    col=fclose (f_cht);
       if (col==EOF)  // EOF – это значение функции fclose при ошибке
                      // При успехе возвращается 0.
            {
                print f("\nФайл f1.dat не закрыт.");
                return 3;
              }
        return 0;
}
```

//конец программы чтения из файла

```
# include <stdio.h>          // Для функции ввода/вывода.
  int main (void)            // Возвращает 0 при успехе.
   {
                             // Данные для записи в файл
     int n=7;
     long int ln=12l;
     short int sn=5;
     float x=1.5e2;
     long double ld=2.0e-3L;
      file *f_zap;           // Указатель на файл для записи
       int col;              // Возвращаемое значение для fclose.
                             // Открываем файл f2.out для записи.
      f_zap=fopen ("f2.out", "w");
       if (f_zap==NULL)
          {
            printf ("\nФайл f2.out для записи не открыт.");
           return 1;
        }
      …
  } // end main ()
```