

# Лекция 2

Алгоритмические языки и  
программирование

# Часть 1

# Стандартные потоки

- В начале выполнения программы автоматически открываются три потока.
- `stdin` (стандартный поток ввода),
- `stdout` (стандартный поток вывода),
- `stderr` (стандартный поток ошибок).

Обычно эти потоки направляются к консоли, но в средах, которые поддерживают перенаправление ввода/вывода, они могут быть перенаправлены операционной системой на другое устройство. (Перенаправление ввода/вывода поддерживается, например, такими операционными системами, как Windows, DOS, UNIX.)

# ctype.h

В заголовочном файле <ctype.h> описаны функции для проверки и изменения символьной информации.

# Функции библиотеки `ctype.h`

- `isalpha(c)`; – возвращает не нулевое значение, если аргумент является буквой
- `isdigit(c)`; – возвращает не нулевое значение, если аргумент является цифрой
- `isspace(c)`; – возвращает не нулевое значение, если аргумент является пробельным(управляющим) СИМВОЛОМ
- (' ', '\t', '\n', '\r' и др.)
- `islower(c)`; – возвращает не нулевое значение, если аргумент является символом в нижнем регистре
- `isupper(c)`; – возвращает не нулевое значение, если аргумент является символом в верхнем регистре
- `toupper(c)`; – переводит символ в верхний регистр
- `tolower(c)`; – переводит символ в нижний регистр

# Ввод символов через getchar()

- Стандартная библиотека `stdio.h` включает несколько функций для чтения и записи одного символа. Простейшие из них — `getchar` и `putchar`.

```
int getchar ( void );
```

```
int putchar( int character );
```

- За одно обращение к `getchar` считывается следующий символ ввода из текстового потока, и этот символ выдается в качестве результата. Так, после выполнения

```
c = getchar ( );
```

переменная `c` содержит очередной символ ввода.

# Вывод символов через putchar()

- Обращение к putchar приводит к печати одного символа. Так,

```
putchar(c);
```

напечатает содержимое целой переменной `c` в виде символа (обычно на экране). Вызовы `putchar` и `printf` могут произвольным образом перемежаться. Вывод будет формироваться в том же порядке, что и обращения к этим функциям.

# Константа EOF

- EOF является индикатором операционной системы, означающим, что данные в источнике закончились. Источниками данных обычно являются файлы и потоки.
- EOF (аббревиатура от end of file — конец файла) — целая константа, определенная в `<stdio.h>`. Какое значение имеет эта константа — неважно, лишь бы оно отличалось от любого из возможных значений типа `char`.
- Для указания терминалу в UNIX и LINUX «EOF» следует воспользоваться комбинацией клавиш `Ctrl+D`. В Windows — `Ctrl+Z`.



# Копирование ввода на вывод 1

```
#include <stdio.h>
int main()
{
    int c;
    c = getchar();
    while (c != EOF)
    {
        putchar (c);
        c = getchar();
    }
    return 0;
}
```

# Копирование ввода на вывод 2

```
#include <stdio.h>
int main()
{
    int c;
    while ((c = getchar()) != EOF)
        putchar (c);
    return 0;
}
```

# Подсчёт символов 1

```
#include <stdio.h>
{
    long int nc = 0;
    while (getchar() != EOF)
        nc++;
    printf ("%ld\n", nc);
    return 0;
}
```

# Подсчёт символов 2

```
#include <stdio.h>
int main()
{
    int nc;
    for (nc = 0; getchar() != EOF; nc++)
        ;
    printf ("%d\n", nc);
    return 0;
}
```

# Лабораторные работы

# Проверка символов

- Напишите программу, которая проверит является ли символ управляющей последовательностью.

# Перевод символов

- Напишите программу, которая переведет символ в верхний регистр.

# Подсчет строк

- Напишите программу, которая считает строки из входного потока.
  
- Примечание:
- Использовать функцию `getchar()`;
- Использовать циклы