

# Аргументы функции main()

```
int main(int argc, char *argv[])
```

- `argc` – количество аргументов в командной строке; первым аргументом считается имя программы, поэтому  $argc \geq 1$
- `*argv[]` – массив указателей каждый элемент которого содержит аргумент командной строки **в текстовом виде**

## Тестовая программа печати аргументов

```
#include <stdio.h>

int main(int argc, char *argv[]) {
    printf("argc==%d\n", argc);
    for ( int i = 0; i < argc; i++ )
        printf("argv[%d] == %s\n", i, argv[i]);
}
```

## Назовем эту программу test

```
./test
```

```
argc==1
```

```
argv[0] == ./test
```

```
./test second example
```

```
argc==3
```

```
argv[0] == ./test
```

```
argv[1] == second
```

```
argv[2] == example
```

```
./test -test 1          23
```

```
argc==4
```

```
argv[0] == ./test
```

```
argv[1] == -test
```

```
argv[2] == 1
```

```
argv[3] == 23
```

# Библиотечные функции `atoi()` и `atof()`

```
#include <stdlib.h>
```

Функции преобразуют строку в число:

- для целых чисел (`ascii to integer`):  
`int atoi(const char *str);`  
функции `atol()` и `atoll()` возвращают `long` и `long long`
- для чисел с плавающей точкой (`ascii to float`):  
`double atof(const char *str);`
- конец числа определяется символом, который не может быть частью числа
- если число выходит за пределы допустимых значений, результат не определен

☞ `atoi()` и `atof()` не проверяют ошибки преобразования

более сложные функции: `strtol()` и `strtod()` проверяют правильность преобразования