

При запуске программы в командной строке может указываться только название исполняемого файла, например:

program.exe

либо название запускаемой программы с указанием дополнительных аргументов, например:

program.exe arg1 arg2 arg3

Для получения строковых значений аргументов командной строки используются параметры, передаваемые в функцию `main`

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

Здесь:

`int` `argc` – (*arguments counter*) целочисленный параметр, значение которого соответствует числу элементов в командной строке, $argc \geq 1$

`char*` `argv[]` – (*arguments values*) массив строк с аргументами командной строки. `argv[0]` – строка с названием программы, `argv[1]` – строка с первым аргументом, `argv[2]` – строка со вторым аргументом, ... , `argv[argc-1]` – строка с последним аргументом командной строки.

Например, для командной строки

```
program.exe file1.txt file2.txt 123
```

```
argc = 4  
argv[0] : "program.exe"  
argv[1] : "file1.txt"  
argv[2] : "file2.txt"  
argv[3] : "123"  
argv[4] : NULL
```

Числовые аргументы могут быть получены из строк с их описанием при помощи функций `atoi()` – для целых чисел типа `int`, `atol()` – для целых чисел типа `long`, `atof()` – для вещественных чисел типа `double`. Для рассмотренного выше примера:

```
atoi( argv[3] ) = atoi( "123" ) → 123
```

Обращение к конкретному i -му символу в строке j -го аргумента может быть осуществлено через указание двух индексов: `argv[j][i]`.

В случае запуска программы с указанием пути до исполняемого файла строка `argv[0]` также будет содержать символы данного пути. Например:

`argv[0] : "i:\program\program.exe"`

/ Пример 1.
Вывод всех аргументов, заданных
в командной строке программы */*

```
#include <stdio.h>
```

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

Список с переменными окружения (переменными среды, *environment variables*) и их значениями в командной строке можно вывести при помощи команды `set` .

Например:

```
ALLUSERSPROFILE=C:\ProgramData
APPDATA=C:\Users\user\AppData\Roaming
CommonProgramFiles=C:\Program Files\Common Files
ComSpec=C:\Windows\system32\cmd.exe
HOMEDRIVE=C:
. . .
```

Для получения доступа к строкам с переменными окружения в функции `main` используется третий из списка её параметров.

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

Здесь:

`char* arge[]` – массив строк с переменными окружения и их значениями

Например:

```
arge[0] : "ALLUSERSPROFILE=C:\ProgramData"  
arge[1] : "APPDATA=C:\Users\user\AppData\Roaming"  
  . . .  
arge[ ] : NULL
```

Обращение к конкретному *i*-му символу в строке *j*-й переменной окружения может быть осуществлено через указание двух индексов: `arge[j][i]`.

* Получить строку со значением переменной окружения по её имени также можно при помощи функции `getenv()`.

/ Пример 2.*

*Вывод всех строк с переменными окружения */*

```
#include <stdio.h>
```

```
int main(int argc, char* argv[], char* arge[])
{
    int i;

    for(i=0; arge[i]!=NULL; i++)
        printf("Env. var. #%d :\t\"%s\"\\n", i, arge[i]);

    return 0;
}
```