

Тема 1.2

Функции и указатели в языке «С»

```
#include <stdio.h>
int main(void)
{
    printf("Hello, World!\n");
    return 0;
}
```

C/C++

```
1 #include <stdlib.h> // подключение заголовка с функцией rand rand()
2 int randomNumber = rand(); // стандартная функция генерации случайных чисел
```

Вот так выглядит вызов стандартной функции

```
1 returnedDataType functionName ( dataType par1, ..., dataType parN );
```

returnedDataType — тип данных, возвращаемого функцией, значения;

functionName — имя функции

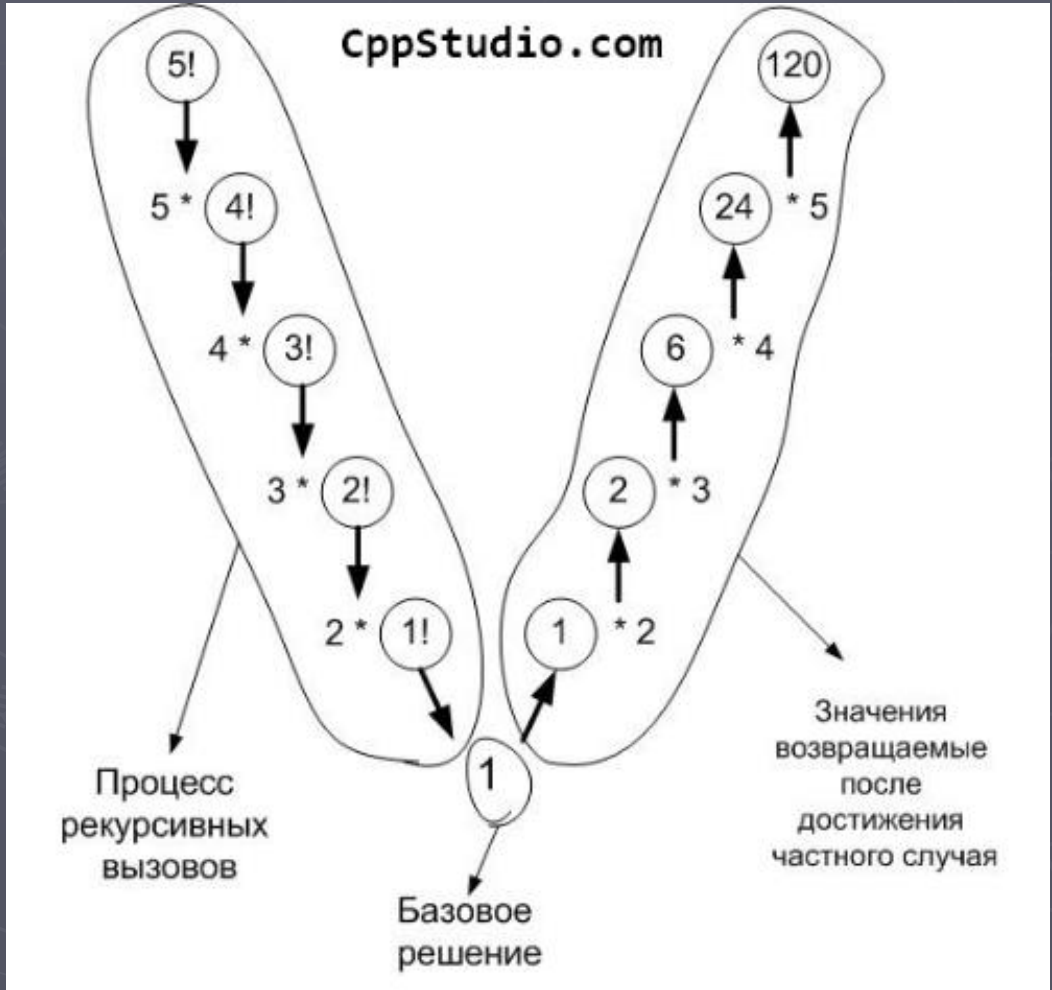
dataType — тип данных параметра функции, это тот же самый тип данных, что и при объявлении переменной

par1 ... parN — параметры функции.

```
1 int mult ( int x, int y );
```

```
1 // factorial.cpp: определяет точку входа для консольного приложения.
2
3 #include "stdafx.h"
4 #include <iostream>
5 using namespace std;
6
7 unsigned long int factorial(unsigned long int); // прототип рекурсивной функции
8 int i = 1; // инициализация глобальной переменной для подсчёта кол-ва рекурсивных вызовов
9 unsigned long int result; // глобальная переменная для хранения возвращаемого результата рекурсивной ф
10
11 int main(int argc, char* argv[])
12 {
13     int n; // локальная переменная для передачи введенного числа с клавиатуры
14     cout << "Enter n!: ";
15     cin >> n;
16     cout << n << "!" << "=" << factorial(n) << endl; // вызов рекурсивной функции
17     system("pause");
18     return 0;
19 }
20
21 unsigned long int factorial(unsigned long int f) // рекурсивная функция для нахождения n!
22 {
23     if (f == 1 || f == 0) // базовое или частное решение
24         return 1; // все мы знаем, что 1!=1 и 0!=1
25     cout << "Step\t" << i << endl;
26     i++; // операция инкремента шага рекурсивных вызовов
27     cout << "Result= " << result << endl;
28     result = f * factorial(f - 1); // функция вызывает саму себя, причём её аргумент уже на 1 меньше
29     return result;
30 }
```

```
CppStudio.com
Enter n!: 5
Step 1
Result= 0
Step 2
Result= 0
Step 3
Result= 0
Step 4
Result= 0
5!=120
```



```
1 //пример использования функции atof
2 #include <iostream>
3 #include <cstdlib>
4 #include <cmath>
5
6 int main()
7 {
8     double pi = 3.1415926535;           // число ПИ
9     char input [256];
10    std::cout << "Введите угол в градусах: ";
11    std::cin >> input ;
12
13    double agl = atof( input );         // преобразуем строку к типу данных float
14
15    std::cout << "Синус угла "
16              << agl << " градусов = "
17              << sin(agl * pi / 180) << "n"; // вычисляем синус угла, переведённого в радианы
18    return 0;
19 }
```

| Функция | Пояснение |
|---------------------------------|---|
| <code>strlen(имя_строки)</code> | определяет длину указанной строки, без учёта нуль-символа |

Копирование строк

| | |
|--------------------------------|--|
| <code>strcpy(s1,s2)</code> | выполняет побайтное копирование символов из строки <code>s2</code> в строку <code>s1</code> |
| <code>strncpy(s1,s2, n)</code> | выполняет побайтное копирование <code>n</code> символов из строки <code>s2</code> в строку <code>s1</code> . возвращает значения <code>s1</code> |

Конкатенация строк

| | |
|-------------------------------|--|
| <code>strcat(s1,s2)</code> | объединяет строку <code>s2</code> со строкой <code>s1</code> . Результат сохраняется в <code>s1</code> |
| <code>strncat(s1,s2,n)</code> | объединяет <code>n</code> символов строки <code>s2</code> со строкой <code>s1</code> . Результат сохраняется в <code>s1</code> |

Сравнение строк

| | |
|--------------------------------|---|
| <code>strcmp(s1,s2)</code> | сравнивает строку <code>s1</code> со строкой <code>s2</code> и возвращает результат типа <code>int</code> : 0 –если строки эквивалентны, >0 – если <code>s1<s2</code> , <0 — если <code>s1>s2</code> С учётом регистра |
| <code>strncmp(s1,s2,n)</code> | сравнивает <code>n</code> символов строки <code>s1</code> со строкой <code>s2</code> и возвращает результат типа <code>int</code> : 0 –если строки эквивалентны, >0 – если <code>s1<s2</code> , <0 — если <code>s1>s2</code> С учётом регистра |
| <code>stricmp(s1,s2)</code> | сравнивает строку <code>s1</code> со строкой <code>s2</code> и возвращает результат типа <code>int</code> : 0 –если строки эквивалентны, >0 – если <code>s1<s2</code> , <0 — если <code>s1>s2</code> Без учёта регистра |
| <code>strnicmp(s1,s2,n)</code> | сравнивает <code>n</code> символов строки <code>s1</code> со строкой <code>s2</code> и возвращает результат типа <code>int</code> : 0 –если строки эквивалентны, >0 – если <code>s1<s2</code> , <0 — если <code>s1>s2</code> Без учёта регистра |

Обработка символов

| | |
|-------------------|---|
| isalnum(c) | возвращает значение <code>true</code> , если <code>c</code> является буквой или цифрой, и <code>false</code> в других случаях |
| isalpha(c) | возвращает значение <code>true</code> , если <code>c</code> является буквой, и <code>false</code> в других случаях |
| isdigit(c) | возвращает значение <code>true</code> , если <code>c</code> является цифрой, и <code>false</code> в других случаях |
| islower(c) | возвращает значение <code>true</code> , если <code>c</code> является буквой нижнего регистра, и <code>false</code> в других случаях |
| isupper(c) | возвращает значение <code>true</code> , если <code>c</code> является буквой верхнего регистра, и <code>false</code> в других случаях |
| isspace(c) | возвращает значение <code>true</code> , если <code>c</code> является пробелом, и <code>false</code> в других случаях |
| toupper(c) | если символ <code>c</code> , является символом нижнего регистра, то функция возвращает преобразованный символ <code>c</code> в верхнем регистре, иначе символ возвращается без изменений. |

Функции поиска

| | |
|-----------------------|---|
| strchr(s,c) | поиск первого вхождения символа <code>c</code> в строке <code>s</code> . В случае удачного поиска возвращает указатель на место первого вхождения символа <code>c</code> . Если символ не найден, то возвращается ноль. |
| strcspn(s1,s2) | определяет длину начального сегмента строки <code>s1</code> , содержащего те символы, которые не входят в строку <code>s2</code> |
| strspn(s1,s2) | возвращает длину начального сегмента строки <code>s1</code> , содержащего только те символы, которые входят в строку <code>s2</code> |
| strprbk(s1,s2) | Возвращает указатель первого вхождения любого символа строки <code>s2</code> в строке <code>s1</code> |

Функции преобразования

| | |
|-----------------|--|
| atof(s1) | преобразует строку <code>s1</code> в тип <code>double</code> |
| atoi(s1) | преобразует строку <code>s1</code> в тип <code>int</code> |
| atol(s1) | преобразует строку <code>s1</code> в тип <code>long int</code> |

Функции стандартной библиотеки ввода/вывода `<stdio>`

| | |
|-------------------|--|
| getchar(c) | считывает символ <code>c</code> со стандартного потока ввода, возвращает символ в формате <code>int</code> |
| gets(s) | считывает поток символов со стандартного устройства ввода в строку <code>s</code> до тех пор, пока не будет нажата клавиша ENTER |

```
int    (*pf)();           // без контроля параметров вызова
int    (*pf)(void);      // без параметров, с контролем по прототипу
int    (*pf)(int, char*); // с контролем по прототипу
```

```
int    INC(int a) { return a+1; }
extern int DEC(int);
int    (*pf)(int);
pf = &INC;
pf = INC;                // присваивание указателя
int    (*pp)(int) = &DEC; // инициализация указателя
```



```
1 // argc_argv.cpp: определяет точку входа для консольного приложения.
2
3 #include "stdafx.h"
4 #include <iostream>
5 using namespace std;
6
7 int main(int argc, char* argv[])
8 {
9     if (argc > 1)// если передаем аргументы, то argc будет больше 1(в зависимости от кол-ва аргумен
10 {
11     cout << argv[1]<<endl;// вывод второй строки из массива указателей на строки(нумерация в
12 } else
13     {
14     cout << "Not arguments" << endl;
15     }
16     system("pause");
17     return 0;
18 }
```

```
CppStudio.com
Microsoft Windows [Version 6.1.7600]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.

C:\Use... \top\argc_argv\Release\argc_argv.exe
Not arguments
Для продолжения нажмите любую клавишу . . .
```

```
CppStudio.com
Microsoft Windows [Version 6.1.7600]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.

C:\Use... \top\argc_argv\Release\argc_argv.exe Open
Open
Для продолжения нажмите любую клавишу . . .
```

```
CppStudio.com
Microsoft Windows [Version 6.1.7600]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.

C:\Use... \top\argc_argv\Release\argc_argv.exe "It work
"
It work
Для продолжения нажмите любую клавишу . . .
```