

Модуль 1 Занятие 4

CS50.h

```
#include <cs50.h>
#include <stdio.h>

int main(void)
{
    int i = get_int("Input: ");
    printf("Output: %i\n", i);
}
```

Эта функция запрашивает у пользователя файл `int`. Если пользователь вводит что-либо, кроме `int` (или значения, которое не может поместиться в `int`), функция снова запрашивает у пользователя.

Эта функция возвращает введенные пользователем данные в виде файла `int`.

Problem set 1

Напишите программу, которая приветствует пользователя, выводя слово **Hello**, введенное имя и знаки препинания по образцу,

```
Введите своё имя: Dmitry  
hello, Dmitry
```

Арифметические операции

Основные арифметические операторы языка Си.

- + оператор сложения
- оператор вычитания
- * оператор умножения
- % оператор взятия остатка от деления
- / оператор деления

Приведение типов

Пример явного приведения типа

```
int x = 5;  
double y = 15.3;  
x = (int) y;  
y = (double) x;
```

Пример неявного приведения типа

```
int x = 5;  
double y = 15.3;  
y = x; //здесь происходит неявное приведение типа к double  
x = y; //здесь происходит неявное приведение типа к int
```

Составные операторы присваивания

| Оператор | Эквивалентное выражение |
|---------------------------------|------------------------------------|
| <code>a += b;</code> | <code>a = a + b;</code> |
| <code>a -= b;</code> | <code>a = a - b;</code> |
| <code>a *= b;</code> | <code>a = a * b;</code> |
| <code>a /= b;</code> | <code>a = a / b;</code> |
| <code>a %= b;</code> | <code>a = a % b;</code> |
| <code>a <<= b;</code> | <code>a = a << b;</code> |
| <code>a >>= b;</code> | <code>a = a >> b;</code> |
| <code>a >>>= b;</code> | <code>a = a >>> b;</code> |
| <code>a &= b;</code> | <code>a = a & b;</code> |
| <code>a ^= b;</code> | <code>a = a ^ b;</code> |
| <code>a = b;</code> | <code>a = a b;</code> |

Условные операторы

```
if ( УСЛОВИЕ  
    (результат выражения) )  
{  
    Инструкция(и)  
}  
else if ( УСЛОВИЕ  
         (результат выражения) )  
{  
    Инструкция(и)  
}  
else  
{  
    Инструкция(и)  
}
```

Условные операторы

```
if (условие) оператор1;  
else оператор2;
```

Если условие истинно, выполняется оператор 1, если ложно, то выполняется оператор2.

Пример 1

```
if (x > y) max = x;  
else  
max = y;
```

Пример 3

```
if (x < 0) x = -x; abs = x;
```

Пример 2

```
if (x < 0) {  
x = -x;  
cout<< "Изменить значение x на  
противоположное по знаку"; }  
abs = x;
```

Пример 4

```
if (x < 0)  
cout<< "Отрицательная величина";  
else if (x > 0) cout<< "Положительная величина";  
else cout<< "Ноль";
```

1. Дано целое число. Если оно является положительным, то прибавить к нему 1; в противном случае не изменять его. Вывести полученное число.
2. Дано целое число. Если оно является положительным, то прибавить к нему 1; в противном случае вычесть из него 2. Вывести полученное число
3. Дано целое число. Если оно является положительным, то прибавить к нему 1; если отрицательным, то вычесть из него 2; если нулевым, то заменить его на 10. Вывести полученное число.

Логические операции

Конъюнкция

(AND)

| a | b | $a \wedge b$ |
|-----|-----|--------------|
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 1 | 1 |

Дизъюнкция

(OR)

| a | b | $a \vee b$ |
|-----|-----|------------|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 1 | 1 |

1 В математике функция $\text{sign}(x)$ (знак числа) определена так:

$\text{sign}(x) = 1$, если $x > 0$,

$\text{sign}(x) = -1$, если $x < 0$,

$\text{sign}(x) = 0$, если $x = 0$.

Для данного числа x выведите значение $\text{sign}(x)$.

2 Дано натуральное число. Требуется определить, является ли год с данным номером високосным. Если год является високосным, то выведите YES, иначе выведите NO. Напомним, что в соответствии с григорианским календарем, год является високосным, если его номер кратен 4, но не кратен 100, а также если он кратен 400.

3 Даны три числа. Найти сумму двух наибольших из них

Домашнее задание

- 1) Шахматный король ходит по горизонтали, вертикали и диагонали, но только на 1 клетку. Даны две различные клетки шахматной доски, определите, может ли король попасть с первой клетки на вторую одним ходом. Программа получает на вход четыре числа от 1 до 8 каждое, задающие номер столбца и номер строки сначала для первой клетки, потом для второй клетки. Программа должна вывести **YES**, если из первой клетки ходом короля можно попасть во вторую или **NO** в противном случае.
- 2) Даны три целых числа. Найти количество положительных чисел в исходном наборе.
- 3) Даны три целых числа. Найти количество положительных и количество отрицательных чисел в исходном наборе.

Модуль 2 Занятие 1

ЦИКЛЫ

- ▶ Циклом называется многократное выполнение одних и тех же действий.
- ▶ Например, чтобы программа сосчитала от 1, скажем, до миллиона, в ней нужно записать цикл.
- ▶ Цикл — это та часть кода программы, которая выполняется заданное количество раз.
- ▶ Цикл имеет три части:
 - начало;
 - средняя часть, которая повторяется;
 - конец.

Цикл While

- ▶ Когда мы не знаем, сколько итераций должен произвести цикл, нам понадобится цикл **while** или **do...while**. Синтаксис цикла **while** в C++ выглядит следующим образом.
- ▶ Данный цикл будет выполняться, пока условие, указанное в круглых скобках является истиной.

```
1 while(условие)
2 {
3     // выполняемые действия
4 }
```

Задачи:

- 1) Программа должна выводить произведение десяти чисел, введённых пользователем с клавиатуры.
- 2) Программа должна выводить среднее арифметическое десяти вещественных чисел, введённых с клавиатуры.

Цикл do while

- ▶ Тело цикла do while всегда выполняется хотя бы один раз. После выполнения тела цикла проверяется условие. Если оно истинно, то выполнение переходит к началу блока do и тело цикла выполняется снова.

```
do  
{  
    тело цикла  
} while (проверка условия);
```

Список задач:

- 1) Организовать непрерывный ввод чисел с клавиатуры, пока пользователь не введёт 0. После ввода нуля, показать на экран количество чисел, которые были введены, их общую сумму и среднее арифметическое. Подсказка: необходимо объявить переменную-счетчик, которая будет считать количество введенных чисел, и переменную, которая будет накапливать общую сумму чисел.
- 2) Необходимо суммировать все нечётные целые числа в диапазоне, который введёт пользователь с клавиатуры.

Самостоятельно

Список задач:

- 1) Вводится целое число с клавиатуры. Необходимо вывести все точные квадраты натуральных чисел, не превосходящие данного числа N .
- 2) Вводится целое число с клавиатуры. Если число меньше нуля вывести строку "Неправильный ввод". Засчитывать, как правильный ввод только те числа, которые больше нуля.
- 3) Необходимо, чтоб программа выводила на экран вот такую последовательность:
7 14 21 28 35 42 49 56 63 70 77 84 91 98
- 4) Необходимо вывести на консоль такую последовательность чисел:
1 2 4 8 16 32 64 128 256 512

Домашнее задание

Список задач:

1. Дано целое число $N (> 0)$. Если оно является степенью числа 3, то вывести True, если не является — вывести False.
2. Даны целые числа K и $N (N > 0)$. Вывести N раз число K .
3. Дано несколько чисел. Вычислите их сумму. Сначала вводите количество чисел N , затем вводится ровно N целых чисел. Какое наименьшее число переменных нужно для решения этой задачи?

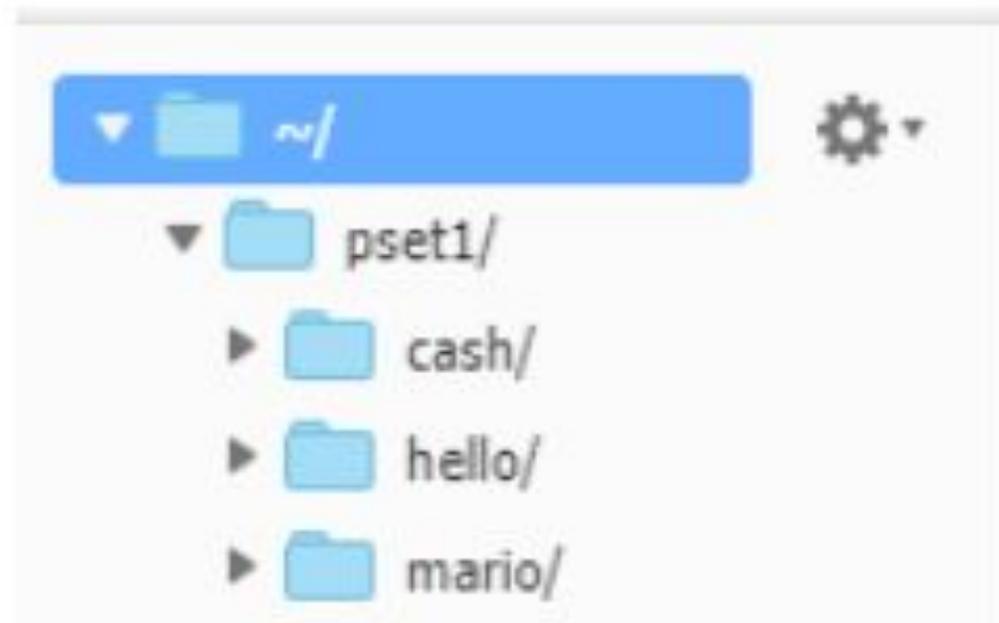
Модуль 2 Занятие 2

Углубление циклов While, do-While

- ▶ Операторы управления циклом:
 - `break` – оператор прекращения цикла. Когда в теле цикла встречается этот оператор, цикл прекращает свою работу
 - `continue` – оператор продолжения цикла. Когда в теле цикла встречается этот оператор, пропускается часть тела цикла, написанная после блока `continue`.

Изучение командной строки

```
comands.txt x +
1 ls - показать все файлы в выбранной папке #list
2 cd <путь до папки> - переместиться в указанную папку #change directory
3 cp <файл для копирования> <путь до папки, в которую хотим скопировать> - копирование файла #copy
4 mv <файл для перемещения> <путь до папки для перемещения> #move
5 mkdir <название директории> - создание новой директории (папки) #make directory
6 rm <название файла> - удаление файла # remove
```



Цикл for

```
1  for (начало; условие; шаг) {  
2    // ... тело цикла ...  
3  }
```

- ▶ Цикл for – параметрический цикл (цикл с фиксированным числом повторений). Для организации такого цикла необходимо осуществить три операции:
 - Инициализация – присваивание параметру цикла начального значения
 - Условие – проверка условия повторения цикла, чаще всего сравнение величины параметра с некоторым граничным значением
 - Модификация – изменение значения параметра для следующего прохождения тела цикла

Задача 1

- ▶ Даны два целых числа A и B . Выведите все числа от A до B включительно, в порядке возрастания, если $A < B$, или в порядке убывания в противном случае

Список задач:

- 1) Даны два целых числа A и B (при этом $A \leq B$). Выведите все числа от A до B включительно.
- 2) Даны два целых числа A и B , $A > B$. Выведите все нечётные числа от A до B включительно, в порядке убывания.

Подготовка к Problem Set 1

Список задач:

- 1) По данному натуральному n вычислите сумму $1^3+2^3+3^3+\dots+n^3$.
- 2) Факториалом числа n называется произведение $1 \times 2 \times \dots \times n$.
Обозначение: $n!$. По данному натуральному n вычислите значение $n!$. Пользоваться математической библиотекой `math` в этой задаче запрещено.
- 3) Дано вещественное число — цена 1 кг конфет. Вывести стоимость 1.2, 1.4, ..., 2 кг конфет.

Домашнее задание

```
$ ./mario
Height: -1
Height: 0
Height: 42
Height: 50
Height: 4
#
##
###
####
```

```
$ ./mario
Height: 8
#
##
###
####
#####
#####
#####
#####
#####
```