**Лабораторная работа №2. Функции.**

***Библиотека функций***

Написать функции согласно варианту. Функции описать в отдельных модулях (файлы ***funcs.h*** и ***funcs.cpp***). Написать программу, которая просит пользователя выбрать одну из функций, запрашивает с клавиатуры необходимые числа и выводит результат выполнения функции на экран.

**Указание:** использовать switch.

1. Проект будет содержать 3 файла:

* заголовочный файл funcs.h, содержащий *объявление* функций;
* файл исходного кода funcs.cpp, содержащий *определение* функций;
* файл исходного кода, содержащий функцию main (main.cpp).

1. В файле main.cpp необходимо подключить заголовочный файл funcs.h:

* #include "funcs.h"

1. функция должна получать значение n и массив чисел соответствующего типа

**Варианты:**

1. **average** – находит вещественное среднее из n целых чисел.
2. **sigma** – находит среднеквадратичное отклонение n вещественных чисел (double)

***Второй функцией во всех вариантах будет вычисление значений плотности атмосферы для любой промежуточной высоты.***

| Высота, H, м | Плотность, кг/м³ |
| --- | --- |
| 0 | 1,2250 |
| 500 | 1,1673 |
| 1000 | 1,1117 |
| 1500 | 1,0581 |
| 2000 | 1,0065 |
| 2500 | 0,9569 |
| 3000 | 0,9093 |
| 4000 | 0,8194 |
| 5000 | 0,7365 |
| 6000 | 0,6601 |
| 7000 | 0,59 |
| 8000 | 0,5258 |
| 9000 | 0,4671 |
| 10000 | 0,4135 |

***Еще одной функцией во всех вариантах будет вычисление чисел Фибоначчи.***

Дополнить программу из первой части. В файлах funcs.h и funcs.cpp реализовать процедуру, выводящую на экран последовательные *n* чисел Фибоначчи, начиная с *k*-го. Функция называется **fibonacci** и имеет 2 целочисленных параметра:

num – количество выводимых чисел,

nBegin – номер числа, с которого начинается вывод.

В функции main() добавить соответствующий пункт в конструкцию switch, в котором у пользователя запрашиваются с клавиатуры числа ***num*** и ***nBegin***, после чего вызывается функция **fibonacci**.

Числа Фибоначчи: 1,1,2,3,5,8,13,…

Числа Фибоначчи вычисляются по формуле:

Fn=Fn-1+Fn-2, F1=1, F2=1, n=2,3,4,…

**Указание:**

1. Использовать цикл **for**.
2. Для реализации процедуры, необходимо объявлять ее с типом void, вывод чисел Фибоначчи внутри функции.

***Дополнительные задания (не обязательно сейчас. Лабораторная 2.1)***

1. Написать функцию, выводящую **побитно** числа типа short int (16 цифр). Функция не возвращает результата (void) и имеет 1 параметр – целое число.
2. Написать функцию, выводящую на экран **побитно** числа типа double (64 цифры). Функция не возвращает результата (void) и имеет 1 параметр (double) – floating point number.

В функции main() добавить соответствующий пункт в конструкцию switch, в котором у пользователя запрашивается число с клавиатуры, после чего вызывается функция побитной печати.

**Еще одно задание: посмотреть как вызывается и как работает функция qsort из :**

<stdlib.h>

**void** **qsort** ( **void** \* first, **size\_t** number, **size\_t** size, **int** (\*comparator ) ( **const** **void** \*, **const** **void** \*) );

Функция qsort выполняет сортировку num элементов массива, на который ссылается указатель first. Для каждого элемента массива устанавливается размер в байтах, который передается через параметр size. Последний параметр функции qsort — указатель comparator на функцию сравнения, которая используется для определения порядка следования элементов в отсортированном массиве.

Алгоритм сортировки используемый этой функцией сравнивает пары значений, путем вызова указанной функции сравнения, с двумя указателями на элементы массива.

Эта функция не возвращает никакого значения, но изменяет содержимое массива, на который указывает first. Таким образом, элементы массива занимают новые места, согласно отсортированному порядку.

Параметры:

first  
Указатель на первый элемент сортируемого массива.

number  
Количество элементов в сортируемом массиве, на который ссылается указатель first.

size  
Размер одного элемента массива в байтах.

comparator  
Функция, которая сравнивает два элемента. Функция должна иметь следующий прототип:

int funccmp( const void \* val1, const void \* val2 );

Функция должна принимать два параметра — указатели на элементы массива, типа void\*. Эти параметры должны быть приведены к определённым типам данных. Возвращаемое значение этой функции должно быть отрицательным, равным нулю или положительным. Если val1 <, == или >, чем val2, функция должна вернуть отрицательное значение, ноль или положительное значение, соответственно.