# Домашнее задание №1

1. Настройка среды разработки
   1. Установка компилятора в составе пакета qt - <https://download.qt.io/archive/qt/4.8/4.8.6/qt-opensource-windows-x86-mingw482-4.8.6-1.exe>
   2. Добавить в системную переменную PATH путь до библиотек QT и исполняемых файлов (gcc) –
      1. Зайти в «мой компьютер» - «свойства» - «Дополнительные параметры системы» - внизу «переменные среды». Снизу есть список «Системные переменные». Найти в нем переменную с названием «Path». В ней будут указаны пути до каталогов через «;». Необходимо ее выбрать и нажать «Изменить». Далее нажать «Создать» и по одиночки ввести туда пути до:
         1. C:\Qt\4.8.5\bin
         2. C:\Qt\4.8.5\lib
         3. C:\Qt\MinGW\bin
         4. C:\Qt\MinGW\lib
         5. C:\Qt\MinGW\include
      2. ЛИБО qt должна сама создать «консоль» с уже определенной переменной path. В поиске искать надо – Qt 4. Тут надо творчески подойти к этому вопросу. Если есть то п. 1.2.1 делать не надо. Все настроено и можно программировать.
   3. Проверить создание exe файла на тестовой программе (gcc main.cpp –o run.exe)далее запуск run.exe:

#include <stdio.h>

int main()

{

printf("HELLO");

return 0;

}

1. Задачи:
   1. Для корректного отображения русского текста необходимо настроить консоль. Надо явно указать кодировку файла, который ты создал. Для windows это cp1251, но при создании файла он может быть создан в кодировке utf8. По этому надо попробовать выполнить в консоли команду chcp 1251. И запустить программу с printf(“Привет”). Если снова появятся кракозябры то выполнить chcp 65001.
   2. При помощи цикла подсчитать факториал числа. Факториал: произведение всех натуральных чисел от 1 до n включительно. Значение n брать из командной строки.

Код для обработки командной строки (ниже программа написано корректно и следует придерживаться такой практики):

#include <stdio.h>

#include <string.h>

/\* считает факториал числа введенного из командной строки

argc - количество параметров

argv - набор параметров с количеством == argc

Возвращает

0 в случае успеха или

-1 в случае ошибки

\*/

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

{

/\*

Значение по умолчанию. Ключевое слово const означает, что

присвоение значения может быть только 1 раз. Если

дальше в коде написать: defaultN = 10; то будет ошибка

Принято все забитый числа, которые логически не должны меняться

Писать именно с const, что бы их потом никто не смог поменять.

Например число Pi = 3.14 будет выглядеть так: const double pi = 3.14;

\*/

const int defaultN = 5;

int n = defaultN;

/\*

Посмотри код, мы много что не проходили, но попробуй разобраться

Вызывать программу - run.exe 144

argc == 2 - 2 т.к. первый параметр это название программы run.exe,

второй параметр это сам аргумент

argv[1] - взять 1-й аргумент командной строки,

а argv[0] - название программы

sscanf - преобразует первый аргумент в целое число, если

это возможно. Если преобразование успешно,

то возврщает 1, если нет то 0

\*/

if (argc == 2)

{

printf("first arg: %s", argv[0]);

printf("\nsecond arg: %s", argv[1]);

if (sscanf(argv[1], "%d", &n) != 1)

{

printf("\nnot int");

return -1;

}

}

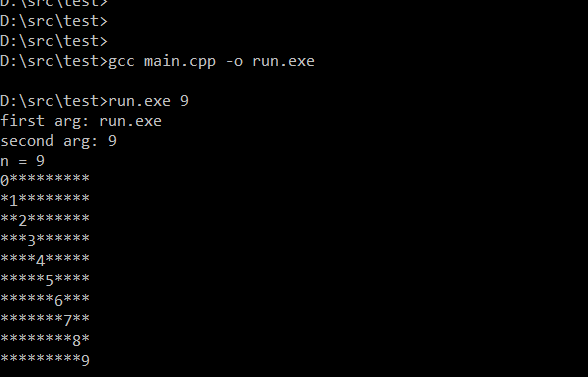
printf("\nn = %d", n);

// твой код

return 0;

}

* 1. Написать программу, которая будет выводить на экран все числа до 200, которые делятся на “N” без остатка. Число “N” брать из командной строки. Можно воспользоваться предыдущим примером. Лучше, если ты его поймешь и попробуешь написать заново сам без подглядок. (не просто запомнить строки, а именно понять, что написано).
  2. Переделать программу в п.2.2. так, что бы она выводила 3 числа введенных с клавиатуры в качестве параметров.
     1. Первый параметр – int
     2. Второй параметр – double (для чтения вместо %d используй %lf)
     3. Третий параметр – int
     4. Если параметров меньше 3, то вывести ошибку «Не хватает параметров»
     5. Если параметров больше 3, то вывести ошибку «Слишком много параметров»
     6. Если переданные параметры не соответствуют типам, то вывести сообщение: «Неверные типы параметров».
     7. Запускать надо будет примерно так: run.exe 5 4.3 14
        1. Т.е. первый параметр целый - 5
        2. Второй с дробной частью – 4.3
        3. Третий целый - 14
  3. Вывести целые числа от 0 до “N” заданное с клавиатуры через параметр, но не больше 20.
     1. Добавить проверку на отрицательное число. Если введено число меньше 0 , то вывести сообщение об ошибке.
     2. Если число больше 9, то вывести 9 чисел по диагонали и на следующей строке вывести предупреждение «Введено больше 9 чисел, остальные будут обрезаны»;
     3. Пустоты в строках заполнить символом «\*»;
     4. Примерно должно получиться что то такое для числа 9:



* 1. Если сделаешь все и будет время еще на пару задач, то пиши, я тебе скину еще задания. Но пока я думаю хватит. Основная задача написать рабочий код с правильными отступами, комментариями (где и что происходит и зачем это надо).