

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

Национальный исследовательский университет ИТМО

МЕГАФАКУЛЬТЕТ ТРАНСЛЯЦИОННЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ПРОГРАММИРОВАНИЯ

ЛАБОРАТОРНАЯ РАБОТА №1

По дисциплине Введение в цифровую культуры и программирования

Название работы

Выполнил: Тарасов Михаил Евгеньевич

Проверил: Страдина Марина Владимировна



Санкт-Петербург, 2020 г.

ЛАБОРАТОРНЫЕ РАБОТЫ. БЛОК – 1.

ВАРИАНТ 20.

Лабораторная работа №1.

Разработайте консольное приложение, обеспечивающее ввод с консоли и вывод на консоль строки, целого числа, вещественного числа или символа в соответствии с номером варианта.

```
#include <stdio.h>

int main(void)
{
    int a;
    char c;

    scanf_s("%c", &c); // Ввод и вывод символа
    printf("%c\n", c);

    scanf_s("%d", &a); // Ввод и вывод целого числа
    printf("%d\n", a);
    return 0;
}
```

Лабораторная работа №2.

Разработайте программу, рассчитывающую значения двух указанных функций. Входные значения аргументов запросите с консоли. Полученные значения функций выведите на консоль.

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

int main(void)
{
    double z1, z2;
    double n, m;

    scanf_s("%lf\n", &n); // Ввод двух чисел m и n
    scanf_s("%lf", &m);

    z1 = (((m - 1) * sqrt(m)) - ((n - 1) * sqrt(n))) / (sqrt(n * pow(m, 3)) + n * m +
    pow(m, 2) - m);
    z2 = (sqrt(m) - sqrt(n)) / (m);
    printf("%lf\n", z1);
    printf("%lf \n", z2); // Вывод чисел (они длжны быть равны)
    return 0;
}
```

Лабораторная работа №3.

Задания

1. Запросите с консоли целое число в указанной системе счисления.
2. Выведите на консоль введённое число в указанной системе счисления.
3. Выведите на консоль введённое в задании 1 число в 16-ричной или 8-ричной системе счисления, а также это же число в той же системе счисления, но сдвинутое влево/вправо на указанное число бит.
4. Выведите на консоль введённое в задании 1 число в 16-ричной или 8-ричной системе счисления (согласно заданию 3), а также это же число в той же системе счисления после применения к нему битовой операции отрицания.
5. Введите с консоли целое число в системе счисления, указанной в задании 3. Выведите на консоль результат указанной битовой операции (и, или, исключающее или) введённого числа и числа, введённого в задании 1 в системе счисления, указанной в задании 3.

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

int main(void)
{
    // Задание 1
    int n;
    scanf_s("%d", &n);
    printf("\n");

    // Задание 2 (вывести в 16-чной системе)
    printf("Task 2\n");
    printf("%x\n", n);
    printf("\n");

    // Задание 3 (вывести в 16-чной системе и сдвиг влево на 4)
    printf("Task 3\n");
    printf("%x\n", n);
    printf("\n");
    printf("%x\n", n << 4);
    printf("\n");

    // Задание 4 (инверсия числа)
    printf("Task 4\n");
    printf("%x\n", n);
    printf("\n");
    printf("%x\n", -n);
    printf("\n");

    // Задание 5
    int n1;
    printf("Task 5\n");
    scanf_s("%x\n", &n1);
    printf("\n");
    printf("%x", n & n1);
    return 0;
}
```

Лабораторная работа №4.

Задания

1. Запросить с консоли целое число и проверить его на попадание в заданный диапазон значений. Результат такой проверки вывести на консоль. При выполнении задания не использовать условный оператор.
2. Запросить с консоли целое число и проверить значение бита с указанным номером в этом числе. Результат такой проверки вывести на консоль. Для удобства разрешается запрашивать указанное число в восьмиричной или шестнадцатиричной системе счисления. При выполнении задания важно обратить внимание на грамотный выбор типа данных анализируемого числа.

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

int main(void)
{
    // Задание 1
    int left_v = 33;
    int right_v = 88;
    int x;
    scanf_s("%d", &x);
    int result = left_v <= x && x <= right_v; //Используем унарную опирацию и
    printf("%d\n", result); // Программа выводит 1 если число находится в нужно диапозоне

    // Задание 2
    int z;
    scanf_s("%d", &z);
    z = z >> 19 ;    // Находим элемент на 20 бите
    printf("%d", z);

    return 0;
}
```