Лабораторная работа №1
Условный оператор, оператор выбора, операторы цикла

Пример 1. Написать программу, запрашивающую целое число с клавиатуры и выводящую строку «нечётное число», если число нечетное и строку «чётное», в противном случае.

```
// Подключаем заголовочный файл для работы функций ввода-вывода информации
#include<stdio.h>
// С функции main() начинается выполнение программы
int main() {
    // Определяем переменную, в которую будет записываться число
    int a;
    // Приглашаем ввести число с клавиатуры
    printf("Введите a: ");
    scanf("%i",&a);
    //Проверяем число на четность и выводим соответствующие сообщения
    if (a % 2 == 1) {  //можно написать просто if(a % 2)- результат не изменится
        printf("нечётное число\n");
    } else {
        printf("чётное число\n");
    }
    //Завершаем программу
    return 0;
}
```

Пример 2. Написать программу, решающую квадратное уравнение. Коэффициенты должны вводиться с клавиатуры.

```
// Подключаем заголовочные файлы для работы функций ввода-
// вывода информации и функции извлечения квадратного корня
#include<stdio.h>
#include<math.h>
// С функции main() начинается выполнение программы
int main() {
    printf("Найдём корни уравнения ax^2+bx+c=0\n");
    // Определяем переменные, в которые будут заноситься значения коэффициентов уравнения
    float a;
    float b;
    float c;
    // Организуем ввод коэффициентов уравнения
    printf("Input a: ");
    scanf("%f", &a);
    printf("Input b: ");
    scanf("%f", &b);
    printf("Input c: ");
    scanf("%f", &c);
    // Определяем переменную для хранения значения дискриминанта
    float D;
    D = b * b - 4 * a * c;
    //Проверяем знак дискриминанта и вычисляем значения корней, если они действительные
    float x1, x2;
    if (D < 0) {
        printf("корни  комплексные\n");
    } else {
        if (D == 0) {
            x1 = -b / (2 * a);
            printf("корень кратности 2: x = %3.4f\n", x1);
        } else {
                x1 = (-b + sqrt(D)) / (2 * a);
                x2 = (-b - sqrt(D)) / (2 * a);
                printf("x1 = %3.4f, x2 = %3.4f\n", x1, x2);
        }
    }
    //Завершаем программу
    return 0;
}
```

Пример 3. Написать программу, возвращающую название дня недели по его номеру.

```
// Подключаем заголовочные файлы для работы функций ввода-вывода информации
#include<stdio.h>
// С  функции main()  начинается выполнение программы
int main() {
    // Определяем переменную, в которую будет заноситься значение номера дня недели
    int n;
    // Организуем ввод номера дня недели
    printf("Введите номер дня недели (1-7): ");
    scanf("%i",&n);
    // Определяем день недели
    switch(n){
        case 1:{ printf("Понедельник\n "); break;}
        case 2:{ printf("Вторник\n "); break;}
        case 3:{ printf("Среда \n"); break;}
        case 4:{ printf("Четверг \n"); break;}
        case 5:{ printf("Пятница \n"); break;}
        case 6:{ printf("Суббота \n"); break;}
        case 7:{ printf("Воскресение \n"); break;}
        default:{ printf("Неточный ввод номера\n"); break;}
    }
    //Завершаем программу
    return 0;
}
```

Пример 4. Написать программу, выводящую на консоль квадраты чисел от 0 до 25.

```
#include<stdio.h>
int main() {
    int a; // очередной элемент последовательности
    // Организуем цикл for, определяя параметр цикла (переменная  int i)
    // внутри самого оператора for;
    for(int i=0; i<=25; i++) {
        a = i*i;
        printf("%i^2 = %i\n",i,a);
    }
    return 0;
}
```

Пример 5. Написать программу, выводящую на консоль первые 6 элементов последовательности, определяемой соотношениями: $a_1 = 2,\ a_{i+1} = a_i^2$.

```
#include<stdio.h>
int main(){
    int a = 2;// переменная инициализирована значением первого элемента
    // Организуем цикл for, определяя параметр цикла (переменная  int i)
    // внутри самого оператора for;
    for (int i=1; i<=6; i++) {
        //выводим текущий элемент последовательности
        printf("%i-ый элемент последовательности = %i\n",i,a);
        a = a*a;//вычисляем следующий элемент последовательности
    }
    return 0;
}
```

Пример 6. Написать программу, вычисляющую сумму первых **n** элементов последовательности, определяемую соотношениями: $a_1 = 2,\ a_{i+1} = a_i^2$.

```
#include<stdio.h>
int main() {
    int a = 2; // первый элемент последовательности
    int s = 0; // переменная для хранения  значения суммы
    int n;     // переменная, в которую будет заноситься количество слагаемых
    // Организуем ввод n
    printf("Введите количество элементов последовательности: ");
    scanf("%i",&n);
    for (int i=1; i<=n; i++){
        a = a*a;// вычисляем следующий элемент последовательности
        s = s+a;// добавляем новый элемент к сумме
    }
    // Выводим значение суммы
    printf("Сумма равна %i\n",s);
    return 0;
}
```

Пример 7. Написать программу, выводящую на консоль положительные целые числа, квадраты которых не превышают некоторого натурального *N*.

```
#include<stdio.h>
int main(){
    int a = 1; // переменная для перебора текущих чисел
    int n;     // переменная, в которую будет заноситься значение границы  N
    // Организуем ввод n
    printf("Введите натуральное число N: ");
    scanf("%i",&n);
    // Организуем цикл while
    while(a*a <= n) {
        printf("Очередное число, чей квадрат меньше числа %i, равно %i\n",n,a);
        a++;
    }
    return 0;
}
```

Пример 8. Написать программу, выводящую на консоль сумму элементов убывающей геометрической прогрессии, превышающих некоторое число *a*.

```
#include<stdio.h>
int main() {
    float b;   // текущий элемент геометрической прогрессии
    float q;   // множитель
    float s;   // текущее значение суммы
    //-------------------------------------------------------------
    printf("Введите первый элемент прогрессии: ");
    scanf("%f",&b);
    printf("Введите знаменатель прогрессии: ");
    scanf("%f",&q);
    //-------------------------------------------------------------
    float a;
    printf("Введите число a: ");
    scanf("%f",&a);
    //-------------------------------------------------------------
    // Организуем цикл do-while
    do {
        b *= q;
        s += b;
    } while(b > a);
    printf("Сумма заданных элементов равна %.4f\n",s);
    return 0;
}
```

Пример 9. Написать программу, запрашивающую целое число до тех пор, пока не будет введено 0 или 1.

```
#include<stdio.h>
int main() {
    int a;
    do {
        printf("Введите целое число: ");
        scanf("%i",&a);
    } while((a != 0)&&(a != 1));
    return 0;
}
```

Задание 0.

Написать программу, вычисляющую значение выражения **f**, не используя стандартную функцию **abs()**.

0.1. f = | x-|x+3| |, **0.2.** f = | x-|x-3|| **0.3.** f = | 2-|x+3|| **0.4.** f = | 2x-|x+3||
0.5. f = | 2x-|x-3|| **0.6.** f = | 5x-|x+3|| **0.7.** f = | 4-|x-3|| **0.8.** f = | |2x+3|-1|

Написать программу, вычисляющую значение выражения **F**.

0.9.
$$F = \begin{cases} 2x^2 + 1, x < 0 \\ x, x > 10 \\ x - 3, else \end{cases}$$

0.10.
$$F = \begin{cases} x^2 - 1, x < 0 \\ x, x > 4 \\ x - 3, else \end{cases}$$

0.11.
$$F = \begin{cases} 2x^2 - 1, x < 0 \\ x - 7, x > 10 \\ x, else \end{cases}$$

0.12.
$$F = \begin{cases} 2x^2, x < 0 \\ x + 2, x > 5 \\ x - 3, else \end{cases}$$

0.13.
$$F = \begin{cases} 2 - x^2, x < 0 \\ x, x > 4 \\ x - 3, else \end{cases}$$

0.14.
$$F = \begin{cases} 2x^2 + 1, x < 0 \\ x + 1, x > 1 \\ x, else \end{cases}$$

0.15.
$$F = \begin{cases} -x^2 + 1, x < 0 \\ 2x, x > 3 \\ x - 3, else \end{cases}$$

Задание 1.

1.1. Даны натуральные числа **a**, **b**. Вычислить произведение **a * b**, используя в программе лишь операции **+**, **-**, **=**, **<**, **>**.
1.2. Даны натуральные числа **a** и **d**. Вычислить частное **q** и остаток **r** при делении **a** на **d**, не используя операций **/** и **%**.
1.3. Дано целое неотрицательное **n**, вычислить **n!** (**0! = 1, n! = n * (n-1)!**).
1.4. Дано целое неотрицательное **n**, вычислить **1/0! + 1/1! + ... + 1/n!**.
1.5. Составить программу, печатающую квадраты всех натуральных чисел от 0 до заданного натурального **n**, но разрешается использовать из арифметических операций лишь сложение и вычитание. Можно использовать **$(a-1)^2 = a^2 - 2a + 1$**.
1.6. Составить программу, печатающую разложение на простые множители заданного натурального числа **n > 0** (другими словами, требуется печатать только простые числа, и произведение напечатанных чисел должно быть равно **n**; если **n = 1**, печатать ничего не надо).
1.7. Разрешено использовать функцию **printf()** для вывода на консоль лишь одной из цифр: 0, 1, 2, ..., 9. Составить программу, печатающую десятичную запись заданного целого числа **n > 0**. *Подсказка*: сначала надо найти степень 10, не превосходящую данное число.
1.8. Разрешено использовать функцию **printf()** для вывода на консоль лишь одной из цифр - 0, 1, 2, ..., 9. Составить программу, печатающую десятичную запись заданного натурального числа **n** , но надо напечатать десятичную запись в обратном порядке. (Для **n = 173** надо напечатать **371**.) *Подсказка*: сначала надо найти степень 10, не превосходящую данное число.
1.9. Написать программу, вычисляющую сумму первых **n** элементов последовательности, элементы которой могут быть получены из формулы **$a_i = (i + a_{i-1} - a_{i-2}) - i^2$**.
1.10. Написать программу, вычисляющую сумму первых **n** элементов последовательности, элементы которой могут быть получены из формулы **$a_i = (i * a_{i-1} + a_{i-2})^2$**.
1.11. Написать программу, вычисляющую сумму первых **n** элементов последовательности, элементы которой могут быть получены из формулы **$a_i = (i + a_{i-1} + i * a_{i-2})^2$**.
1.12. Написать программу, вычисляющую сумму первых **n** элементов последовательности, элементы которой могут быть получены из формулы **$a_i = 1 / (i + a_{i-1} + i)$**.
1.13. Написать программу, вычисляющую сумму первых **n** элементов последовательности, элементы которой могут быть получены из формулы **$a_i = 1 / (i + a_{i-1} + a_{i-2})$**.
1.14. Написать программу, вычисляющую сумму первых **n** элементов последовательности, элементы которой могут быть получены из формулы **$a_i = (i + a_{i-1} / a_{i-2})^2$**.
1.15. Написать программу, вычисляющую сумму первых **n** элементов последовательности, элементы которой могут быть получены из формулы **$a_i = i / a_{i-1} + i * a_{i-2}$**.

1.16. Написать программу, вычисляющую сумму первых **n** элементов последовательности, элементы которой могут быть получены из формулы $a_i = i / a_{i-1} + i * a_{i-2}$.

1.17. Написать программу, вычисляющую сумму первых **n** элементов последовательности, элементы которой могут быть получены из формулы $a_i = i / (a_{i-1} + a_{i-2})$.

1.18. Написать программу, вычисляющую сумму первых **n** элементов последовательности, элементы которой могут быть получены из формулы $a_i = (1 + i * a_{i-2})^2$.

1.19. Написать программу, вычисляющую сумму первых **n** элементов последовательности, элементы которой могут быть получены из формулы $a_i = 1 / (i + a_{i-2})$.

1.20. Написать программу, вычисляющую сумму первых **n** элементов последовательности, элементы которой могут быть получены из формулы $a_i = (2 + a_{i-1} + i + a_{i-2})^2$.

1.21. Написать программу, вычисляющую сумму первых **n** элементов последовательности, элементы которой могут быть получены из формулы $a_i=(2 * i + a_{i-2})^2$.

1.22. Написать программу, вычисляющую сумму первых **n** элементов последовательности, элементы которой могут быть получены из формулы $a_i = (i + a_{i-1} * i * a_{i-2})^2$.

1.23. Написать программу, вычисляющую сумму первых **n** элементов последовательности, элементы которой могут быть получены из формулы $a_i = i^2 + a_{i-1} + a_{i-2}$.