Балтийский государственный технический университет «Военмех»  
им. Д.Ф. Устинова

Кафедра И5

«Информационные системы и программная инженерия»

Индивидуальное домашнее задание №1  
дисциплина: «Основы программирования»  
тема: «Операторы языка Си»

Выполнил: студент гр. ЗИ501 Житенев А.В.

Проверил:

Палехова О.А.

Санкт-Петербург  
2020 г.

Задача 1. Вычислить значение функции

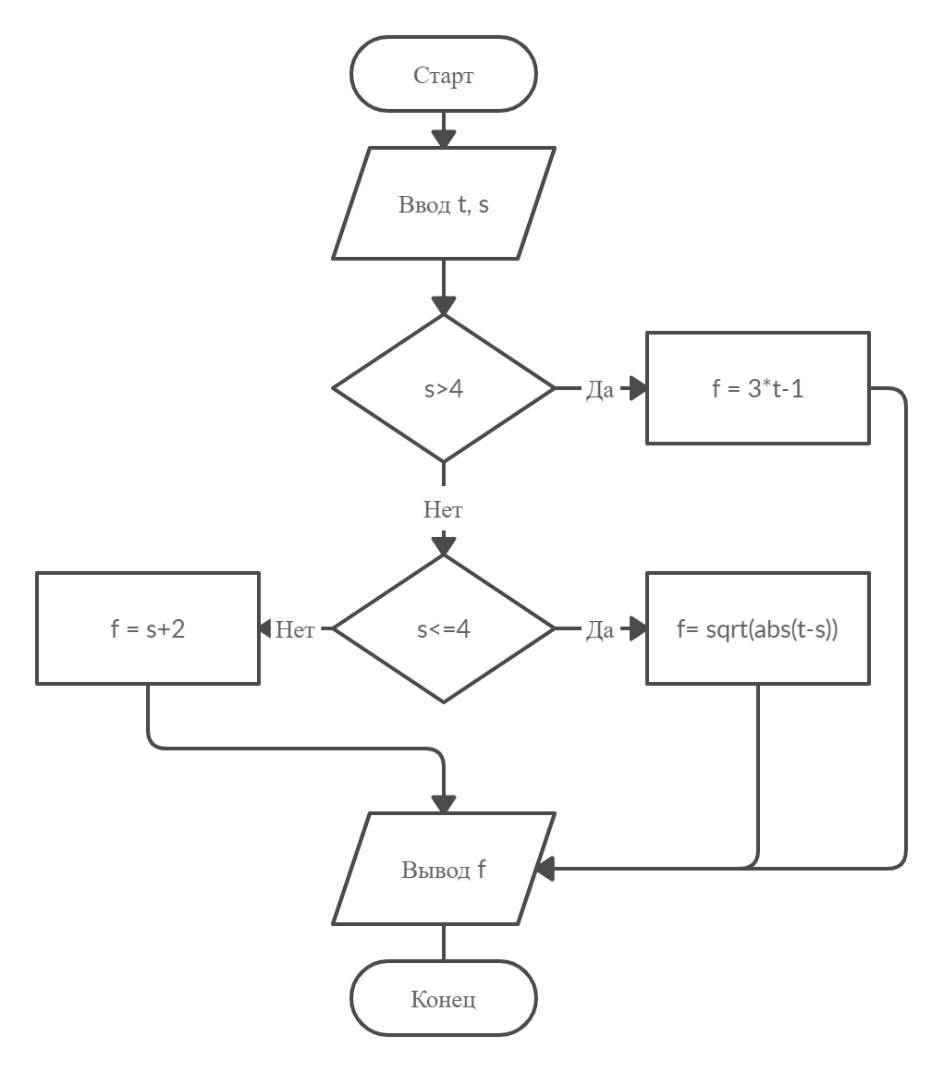
*Исходные данные:*

Аргументы функции *t* и *s.* Так как значения *t* и s могут быть любыми, объявим соответствующие переменные типа *double.*

*Результирующие данные:*

Значение функции *f* , соответствующая переменная тоже будет типа *double.*

*Схема программы*



*Текст программы*

#include <stdio.h>

#include <math.h>

void main(void) {

double s, t, f = 0;

printf("Vvedite s t\n");

scanf\_s("%lf %lf", &s, &t); //получаем аргументы

(s > 4) ? (f = (3 \* t) - 1) : (s <= 4 && t != 0 ? (f = sqrt(fabs(t - s))) : f = s + 2); //вычисляем f

printf("f = %f", f); //вывод результата

return;

}

*Результаты тестирования*

|  |  |  |
| --- | --- | --- |
| **Исходные данные** | **Ожидаемый результат** | **Результат работы программы** |
| s=5, t=1 | 2 | f=2.000000 |
| s=4, t=5 | 1 | f=1.000000 |
| s=4, t=0 | 6 | f=6.000000 |

Задача 2. Вычислить значение функции

*Исходные данные:*

Аргументы функции x, a и b - типа double.

*Результирующие данные:*

Значение функции - переменная *B* типа double.

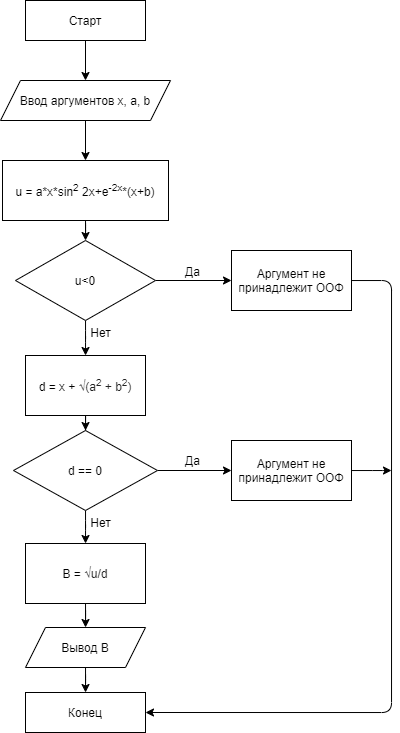
*Предварительные вычисления:*

Чтобы можно было вычислить значение функции, должны быть выполнены следующие условия: неотрицательное подкоренное выражение числителя, ненулевой знаменатель.

*Вспомогательные переменные:*

*u -* подкоренное выражение числителя, d *-* знаменатель дроби; обе переменные типа double.

*Схема программы:*

**

*Текст программы*

#include <stdio.h>

#include <math.h>

void main(void) {

double B, x, a, b;

printf\_s("Vvedite argumenty x a b \n");

scanf\_s("%lf %lf %lf", &x, &a, &b); // получаем аргументы функции

double u = a\*x\*pow(sin(2 \* x), 2) + exp(-2 \* x)\*(x + b); //подкоренное выражение числителя

if (u<0) printf("Argument ne prinadlezhit OOF.\nPodkorennoe vyrazhenie ne mozhet byt' men'she nulya");

else {

double d = x + sqrt(pow(a, 2) + pow(b, 2)); //значение знаменателя

if (!d) printf("Argument ne prinadlezhit OOF.\nZnamenatel' drobi ne mozhet byt' raven nulyu");

else {

B = sqrt(u) / d; // значение функции

printf("B = %lf", B);

}

}

return;

}

*Результаты тестирования*

|  |  |  |
| --- | --- | --- |
| **Исходные данные** | **Ожидаемый результат** | **Результат работы программы** |
| х = 1, a = 1, b = 1 | 0.433935 | B = 0.433935 |
| х = 1, a = -1, b = 1 | Сообщение об ошибке  (отриц. под корнем) | Аргумент не принадлежит ООФ. Подкоренное выражение не может быть меньше нуля |
| x = 0, a = 0, b = 0 | Сообщ. об ошибке  (ноль в числителе) | Аргумент не принадлежит ООФ. Знаменатель дроби не может быть равен нулю |

Задача 3. Дано натуральное трехзначное число. Если все цифры в нем одинаковы, то

оставить заданное число без изменения; если все цифры в нем разные, то меньшую

из них заменить в заданном числе нулём; если две цифры в числе одинаковые, то

получить число с обратным порядком цифр.

*Исходные данные:*

Натуральное число N, тип переменной unsigned int.

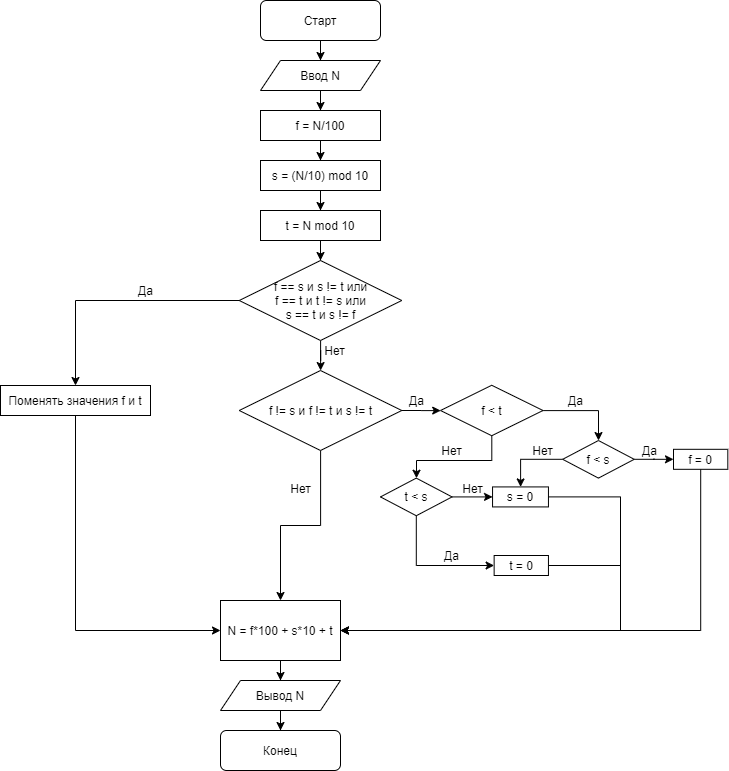
*Результирующие данные:*

Измененное число. Результат записывается в ту же переменную N.

*Вспомогательные переменные:*

Каждую цифру числа будем заносить в отдельную переменную, поэтому определим переменные *f*, s и *t* типа unsigned short.

*Схема программы*

**

*Текст программы*

#include <stdio.h>

void main(void) {

unsigned int N;

printf\_s("Vvedite N \n");

scanf\_s("%d", &N);

unsigned short f = N / 100, s = (N / 10) % 10, t = N % 10;

if ((f == s && s != t) || (f == t && t != s) || (s == t && s != f)) { //если две цифры совпадают

int tmp = f; f = t; t = tmp; //меняем местами первую и вторую цифры

}

else if (f != s && f != t && s != t) { //если все цифры разные

if (f < t) {

if (f < s) f = 0;

else s = 0;

}

else {

if (t < s) t = 0;

else s = 0;

}

}

N = f \* 100 + s \* 10 + t; //собираем число

printf("%i", N); //выводим

return;

}

Результаты тестирования

|  |  |  |
| --- | --- | --- |
| **Исходные данные** | **Ожидаемый результат** | **Результат работы программы** |
| n=777 | 77 | n=777 |
| n=421 | 420 | n=420 |
| n=223 | 322 | n=322 |

Задача 4. Дано натуральное число N. Вычислить произведение

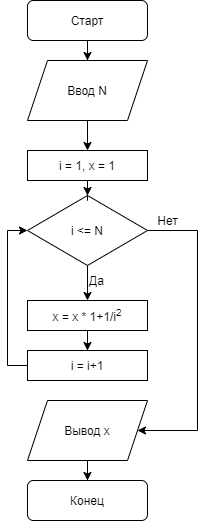
*Исходные данные:*

Натуральное число N – типа unsigned int

*Результирующие данные:*

Переменная x типа double.

*Схема программы:*



*Текст программы*

#include <stdio.h>

void main(void) {

unsigned int N;

double x = 1;

printf\_s("Vvedite N \n");

scanf\_s("%d", &N);

for (int i = 1; i <= N; i++) {

x \*= 1 + 1 / (double)(i\*i);

}

printf("X = %lf", x);

}

Результаты тестирования

|  |  |  |
| --- | --- | --- |
| **Исходные данные** | **Ожидаемый результат** | **Результат работы программы** |
| 1 | 2 | 2 |
| 2 | 2.5 | 2.500000 |
| 5 | 3.069444 | 3.069444 |

Задача 5. Даны натуральные числа a и b. Определить все числа, кратные a и b, меньшие a\*b

(a и b должны быть больше 10).

*Исходные данные:*

Два натуральных числа, тип переменных *a* и *b* unsigned int.

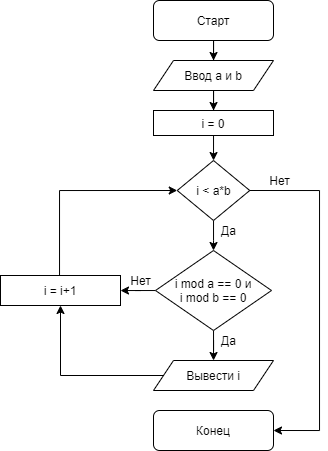
*Результирующие данные:*

Результаты сразу выводятся. Переменных не требуется.

*Вспомогательные переменные:*

Переменная i типа int – для перебора значений.

*Схема программы:*



*Текст программы*

#include <stdio.h>

void main(void) {

unsigned int a, b;

printf\_s("Vvedite a b \n");

scanf\_s("%d %d", &a, &b);

int i = 0;

while (i<a\*b) {

if (i%a == 0 && i%b == 0) printf("%d\n", i);

i++;

}

return;

}

Результаты тестирования

|  |  |  |
| --- | --- | --- |
| **Исходные данные** | **Ожидаемый результат** | **Результат работы программы** |
| а=10 b=11 | Нет результатов | Нет результатов |
| а=12 b=20 | 60 120 180 | 60 120 180 |

Задача 6. Факториал некоторого числа равен p. Найти это число

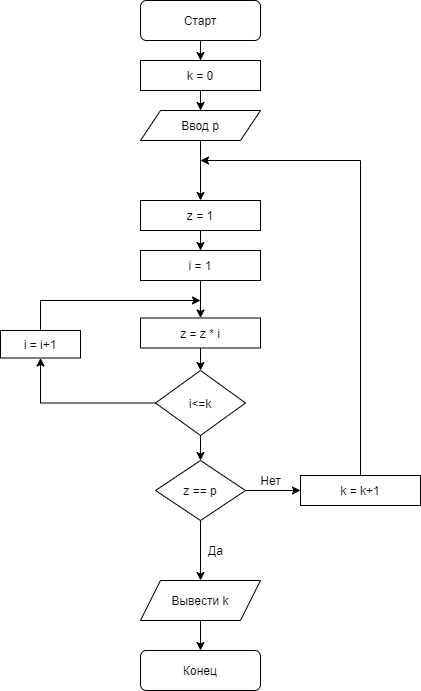
*Исходные данные:*

Число p – типа int.

*Результирующие данные:*

Переменная *k* типа int.

*Схема программы*

**

*Текст программы*

#include <stdio.h>

void main(void) {

int p, k = 0;

printf\_s("Vvedite p \n");

scanf\_s("%d", &p);

while (true) {

int z = 1;

for (int i = 1; i <= k; i++) z \*= i;

if (z == p) break;

else k++;

}

printf("%d", k);

return;

}

Результаты тестирования

|  |  |  |
| --- | --- | --- |
| **Исходные данные** | **Ожидаемый результат** | **Результат работы программы** |
| р=120 | 5 | 5 |
| p= 6227020800 | 13 | 13 |

Задача 7. Вычислить значение суммы бесконечного ряда с заданой точностью ε = 10-4 и значение функции (для проверки)

*Исходные данные:*

Исходный данных нет

*Результирующие данные:*

Значение суммы *s* типа double.

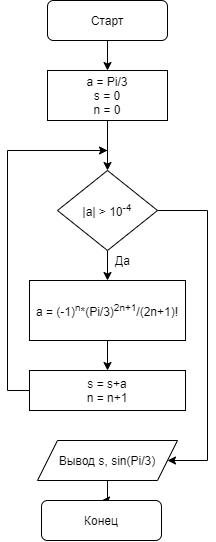
*Вспомогательные переменные:*

*a = PI-3.*

*Предварительные функции:*

*Функция вычисления факториала числа factorial(n)*

*Схема программы*



*Текст программы*

#include <stdio.h>

#define \_USE\_MATH\_DEFINES

#include <math.h>

#define EPS 1e-4

int factorial(int n) {

int r;

for (r = 1; n > 1; r \*= (n--));

return r;

}

void main(void) {

double a = M\_PI / 3;

double s = 0;

int n = 0;

while (fabs(a)>EPS) {

a = pow(-1, n) \* pow(M\_PI / 3, 2 \* n + 1) / factorial(2 \* n + 1);

s += a;

n++;

}

printf("%lf\n", s);

return;

}

Результаты тестирования

|  |  |
| --- | --- |
| **Ожидаемый результат** | **Результат работы программы** |
| 0.866025 | 0.866025 |

Задача 8. Вычислить значение суммы бесконечного ряда с заданной точностью ε=10-6 и значение функции (для проверки) , учесть -1<x≤1

*Исходные данные:*

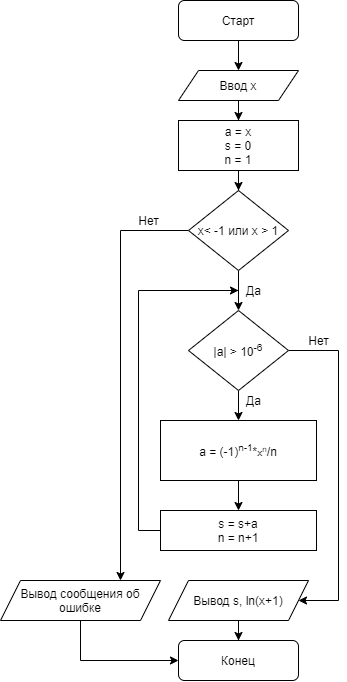
Тип аргумента в задании не указан, поэтому переменная *х* будет типа double. *Результирующие данные:*

Значение суммы *s* тоже будет типа double.

*Вспомогательные переменные:*

*n* - индекс слагаемого - целое число типа int, *a* - значение текущего слагаемого - вещественное число типа double.

*Схема программы*

**

*Текст программы*

#include <stdio.h>

#define \_USE\_MATH\_DEFINES

#include <math.h>

#define EPS 1e-6

void main(void) {

double x;

printf("Vvedite x\n");

scanf\_s("%lf", &x);

double s = 0;

double a = x;

int n = 1;

if (x < -1 || x > 1) {

printf("x is out of range");

}

else {

while (fabs(a) > EPS) {

a = pow(-1, n - 1)\*pow(x, n) / n;

s += a;

n++;

}

printf("%lf\n", s);

printf("%lf\n", log(1 + x));

}

return;

}

Результаты тестирования

|  |  |  |
| --- | --- | --- |
| **Исходные данные** | **Ожидаемый результат** | **Результат работы программы** |
| х=1 | 0.693147 | 0.693147 |
| x=0 | 0 | 0 |
| x=5 | Недопустимое значение | x is out of range |

Задача 9. Даны натуральные числа а, b (а<b). Получить все простые числа p,

удовлетворяющие неравенству a<p<b.

*Исходные данные:*

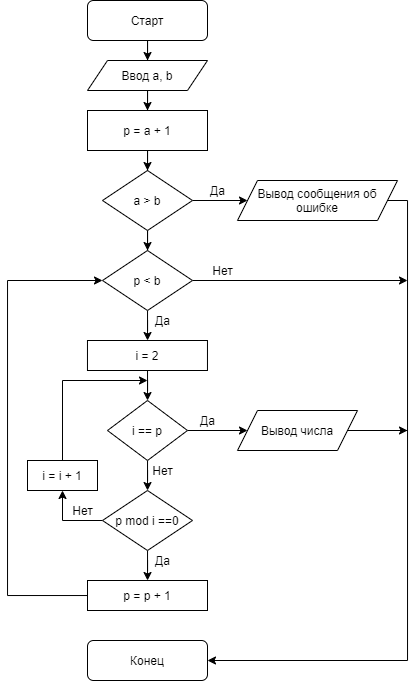
Натуральные числа *a* и *b* типа unsigned int.

*Результирующие данные:*

номер члена последовательности - целое число - переменная *i* типа int; значение

члена последовательности - вещественное число *p* типа int.

*Схема программы*

**

*Текст программы*

#include <stdio.h>

#include <math.h>

void main(void) {

unsigned int a, b;

printf("Vvedite a b\n");

scanf\_s("%d %d", &a, &b);

int p = a + 1;

if (a >= b) printf("a should be < b");

else {

for (p; p < b; p++) { //перебор значение от a до b

for (int i = 2; i <= p; i++) { //проверка на простоту

if (i == p) printf("%i ", p);

if (p%i == 0) break;

}

}

}

return;

}

*Результаты тестирования*

|  |  |  |
| --- | --- | --- |
| **Исходные данные** | **Ожидаемый результат** | **Результат работы программы** |
| a=0, b=0 | Сообщение об ошибке | a should be < b |
| a=1, b=5 | 2, 3 | 2, 3 |
| a=8, b=10 |  |  |