

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное образовательное учреждение высшего
образования**

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

**ОТЧЕТ О ВЫПОЛНЕНИИ
ЛАБОРАТОРНОЙ РАБОТЕ № 1**

«Линейные процессы обработки данных»

студента 2 курса, ИСП-208 группы

Шарибжанов Рамис Фяритович

Направление 09.02.07 – «Информационные системы и
программирование»

Руководитель:
Преподаватель
М.В. Валеев

Работа защищена
« _____ »
“ ” _____ 2021г.

Москва 2021

Оглавление

1. Номер первый	4
1.1. Постановка задачи	4
1.2. Структурные требования	4
1.3. Описание работы	4
1.4.1. Использование переменных	4
1.4.2. Псевдокод программы	4
1.4.3. Блок-схема алгоритма	5
1.4.4. Листинг программной реализации	5
1.4.5. Результаты	6
1.4.6. Тесты	6
2. Номер второй	7
2.1. Постановка задачи	7
2.2. Структурные требования	7
2.3. Описание работы	7
2.4.1. Использование переменных	7
2.4.2. Псевдокод программы	7
2.4.3. Блок-схема алгоритма	7
2.4.4. Листинг программной реализации	9
2.4.5. Результаты	9
2.4.6. Тесты	9
3. Номер третий	10
3.1. Постановка задачи	10
3.2. Структурные требования	10
3.3. Описание работы	10
3.4.1. Используемые переменные	10
3.4.2. Псевдокод программы	10
3.4.3. Блок-схема алгоритма	11
3.4.4. Листинг программной реализации	12
3.4.5. Результат	13
3.4.6. Тесты	13
4. Номер четвертый	14

4.1. Постановка задачи	14
4.2. Структурные требования	14
4.3. Описание работы	14
4.4.1. Используемые переменные	14
4.4.2. Псевдокод программы	14
4.4.3. Блок-схема алгоритма	14
4.4.4. Листинг программной реализации	16
4.4.5. Результат	16
4.4.6. Тесты	16
5. Номер пятый	17
5.1. Постановка задачи	17
5.2. Структурные требования	17
5.3. Описание работы	17
5.5.1. Используемые переменные	17
5.5.2. Псевдокод программы	17
5.5.3. Блок-схема алгоритма	17
5.5.4. Листинг программной реализации	19
5.5.5. Результат	19
5.5.6. Тесты	19
6. СПИСОК ЛИТЕРАТУРЫ	20

1. Номер первый

1.1. Постановка задачи

Вычислите значение функции $y = \frac{e^{-x_1} + e^{-x_2}}{2}$ с клавиатуры в вести x_1 и x_2 и вывести y .

1.2. Структурные требования

Структурных требований нет.

1.3. Описание работы

1.4.1. Использование переменных

X_1 – ввести с клавиатуры

X_2 – ввести с клавиатуры

Y – вывести на экран

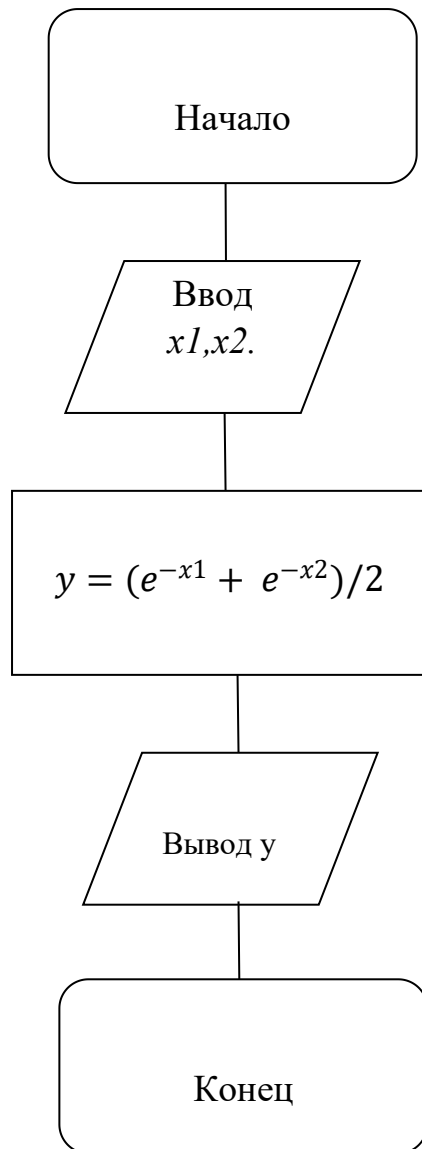
1.4.2. Псевдокод программы

Ввести с клавиатуры x_1 и x_2

Найти значение функции $y = \frac{e^{-x_1} + e^{-x_2}}{2}$

Вывести y

1.4.3. Блок-схема алгоритма

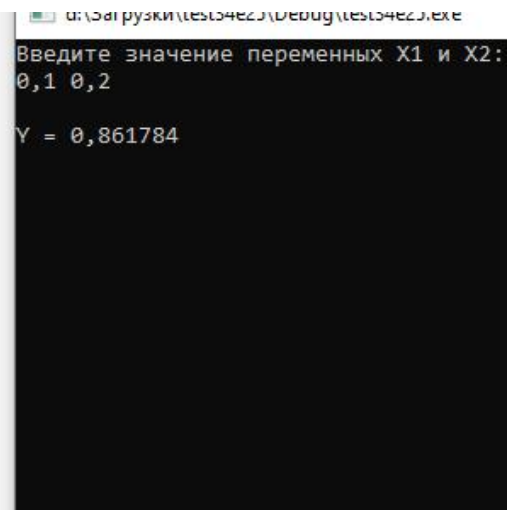


1.4.4. Листинг программной реализации

```
#include "stdafx.h"
#include <stdio.h>
#include <conio.h>
#include <locale>
#include <math.h>
int main()
{
    setlocale(0, "rus");
    float x1, x2, y;
    printf("Введите значение переменных X1 и X2:\n");
    scanf_s("%f%f", &x1, &x2);
    y = (exp(-x1) + exp(-x2))/2;
    printf("\nY = %f", y);
    getch();
    return 0;
}
```

1.4.5. Результаты

```
#include "stdafx.h"
#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <math.h>
int main()
{
    setlocale(0, "rus");
    float x1, x2, y;
    printf("Введите значение переменных X1 и X2:\n");
    scanf_s("%f%f", &x1, &x2);
    y = (exp(-x1) + exp(-x2))/2;
    printf("\nY = %f", y);
    getch();
    return 0;
}
```



D:\31ruski\test34e23\Debug\test34e23.exe
Введите значение переменных X1 и X2:
0,1 0,2
Y = 0,861784

1.4.6. Тесты

X1	X2	Ручной подсчёт	Программный подсчёт
0,1	0,2	0,861784	0,861784
0,23	0,3	0,767676	0,767676
0,45	0,45	0,637628	0,637628
0,12	2	0,511128	0,511128

2. Номер второй

2.1. Постановка задачи

Вычислить площадь треугольника со сторонами a , b , c по формуле Герона $s = \sqrt{p} * (p - a) * (p - b) * (p - c)$ $p = (a + b + c)/2$ клавиатуры в вести a , b , c , p и вывести p и s .

2.2. Структурные требования

Сумма 2 сторон должно быть больше третьего.

a , b , c не должны равняться нулю.

2.3. Описание работы

2.4.1. Использование переменных

a – сторона треугольника

b – сторона треугольника

c – сторона треугольника

p – аргумент для полупериметра треугольника

s – ответ решаемой по формуле Герона

2.4.2. Псевдокод программы

Ввести с клавиатуры a , b , c .

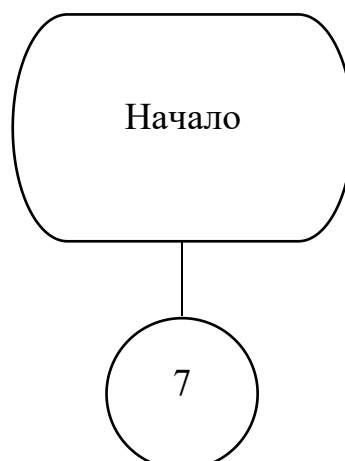
Проверяем чтобы сумма двух чисел было больше третьего числа

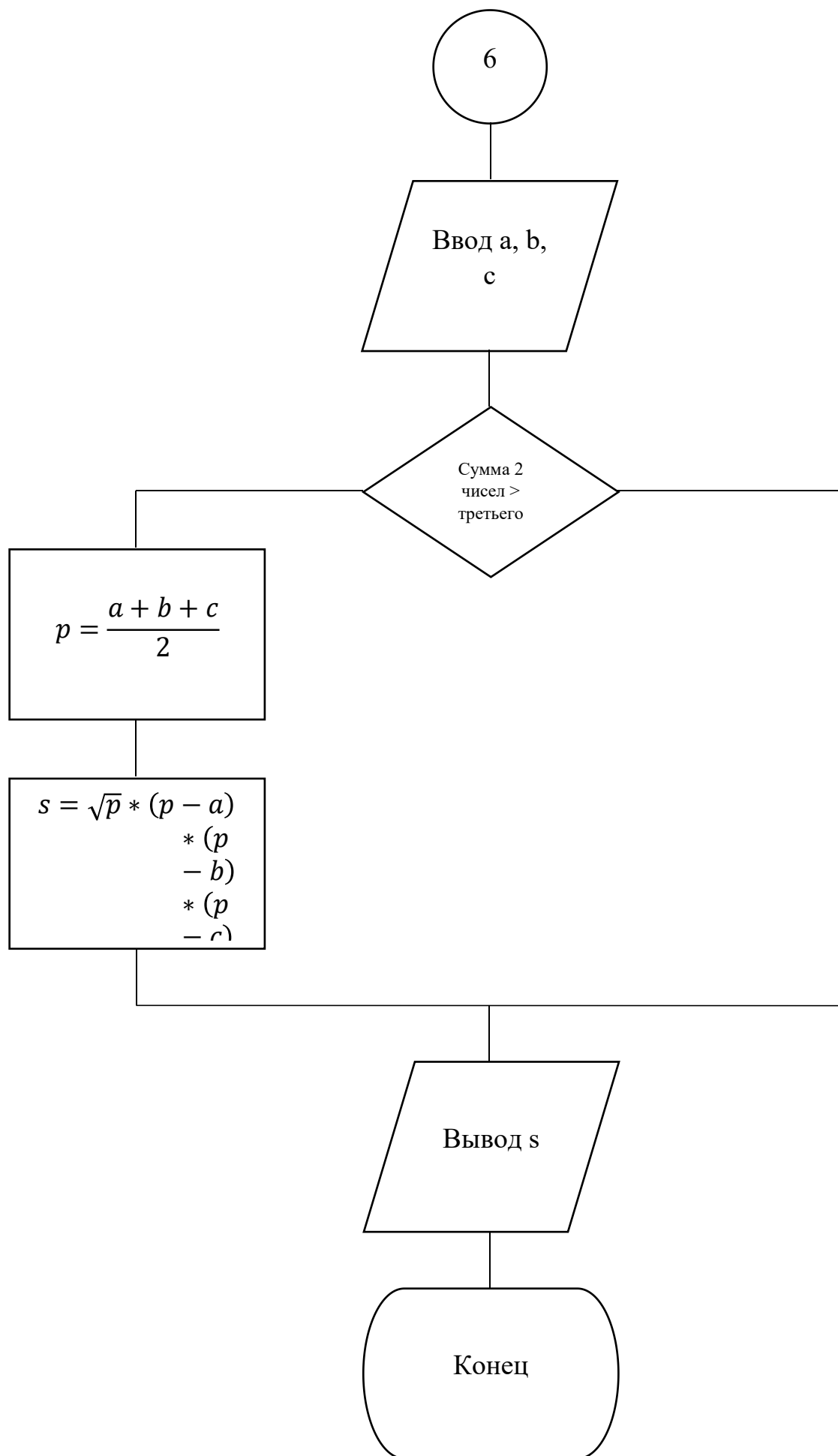
Находим p формуле $p = \frac{a+b+c}{2}$

Находим s $s = \sqrt{p} * (p - a) * (p - b) * (p - c)$

Выводим s

2.4.3. Блок-схема алгоритма



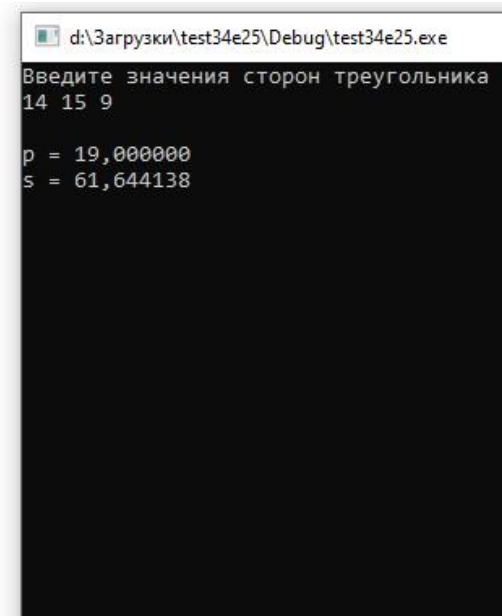


2.4.4. Листинг программной реализации

```
#include "stdafx.h"
#include <stdio.h>
#include <conio.h>
#include <locale>
#include <math.h>
int main()
{
    setlocale(0, "rus");
    float a, b, c, s, p;
    printf("Введите значения сторон треугольника\n");
    scanf_s("%f%f%f", &a, &b, &c);
    if(a>0 && b>0 && c>0 && a+b>c && a+c>b && b+c>a)
    {
        p = (a + b + c)/2;
        s = sqrt(p * (p - a) * (p - b) * (p - c));
        printf("\np = %f", p);
        printf("\ns = %f", s);
    }
    else printf("Такой треугольник не существует\n");
    getch();
    return 0;
}
```

2.4.5. Результаты

```
#include "stdafx.h"
#include <stdio.h>
#include <conio.h>
#include <locale>
#include <math.h>
int main()
{
    setlocale(0, "rus");
    float a, b, c, s, p;
    printf("Введите значения сторон треугольника\n");
    scanf_s("%f%f%f", &a, &b, &c);
    if(a>0 && b>0 && c>0 && a+b>c && a+c>b && b+c>a)
    {
        p = (a + b + c)/2;
        s = sqrt(p * (p - a) * (p - b) * (p - c));
        printf("\np = %f", p);
        printf("\ns = %f", s);
    }
    else printf("Такой треугольник не существует\n");
    getch();
    return 0;
}
```



2.4.6. Тесты

a	b	c	Ручной подсчёт	Программный подсчёт
17	7	18	19	61,644138
34	54	28	58	408,705261
45	45	32	61	672,951721
25	25	25	37,500000	70,632935

3. Номер третий

3.1. Постановка задачи

Вычислить высоты треугольника со сторонами a, b, c по формулам.

$$s = \sqrt{p * (p - a) * (p - b) * (p - c)},$$

$$p = (a + b + c)/2,$$

$$h_a = 2 * s/a,$$

$$h_b = 2 * s/b,$$

$$h_c = 2 * s/c.$$

3.2. Структурные требования

Сумма 2 сторон должно быть больше третьего.

a, b, c не должны равняться нулю.

3.3. Описание работы

3.4.1. Используемые переменные

a – сторона треугольника

b – сторона треугольника

c – сторона треугольника

p – аргумент для полупериметра треугольника

s – ответ решаемой по формуле Герона

ha – высота a

hb – высота b

hc – высота c

3.4.2. Псевдокод программы

Проверяем чтобы сумма двух чисел было больше третьего числа

Находим p формуле $p = \frac{a+b+c}{2}$

Находим s $s = \sqrt{p * (p - a) * (p - b) * (p - c)}$

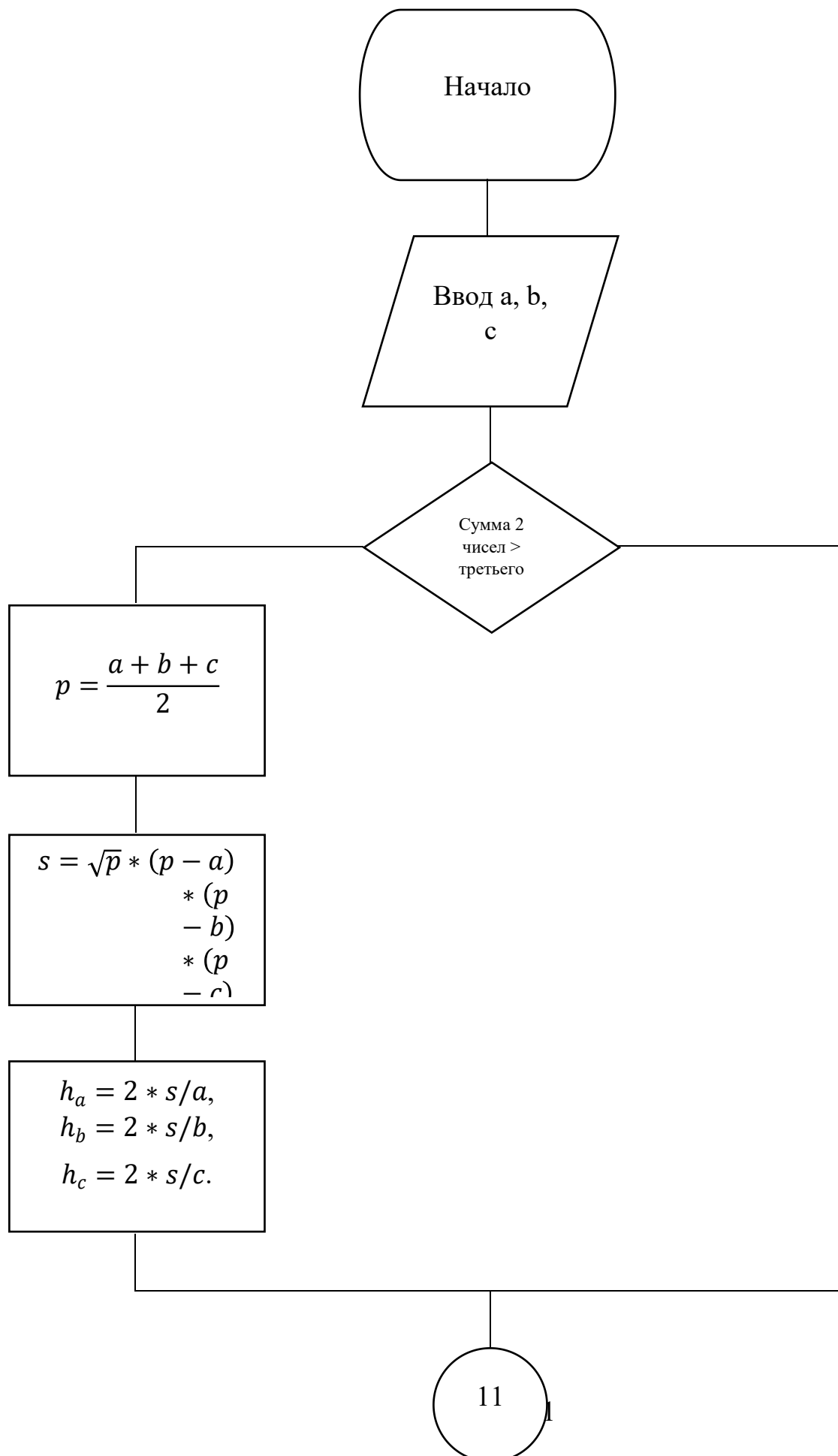
Находим ha по формуле $ha = 2 * s/a$

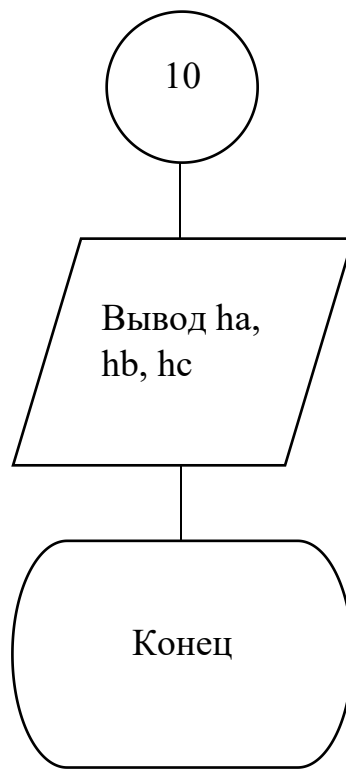
Находим hb по формуле $hb = 2 * s/b$

Находим hc по формуле $hc = 2 * s/c$

Выводим ha, hb, hc

3.4.3. Блок-схема алгоритма





3.4.4. Листинг программной реализации

```
#include "stdafx.h"
#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <math.h>
int main()
{
    setlocale(0, "rus");
    float a, b, c, s, p, h1, h2, h3;
    printf("Введите значения сторон треугольника\n");
    scanf_s("%f%f%f", &a, &b, &c);
    if(a>0 && b>0 && c>0 && a+b>c && a+c>b && b+c>a)
    {
        p = (a + b + c)/2;
        s = sqrt(p * (p - a) * (p - b) * (p - c));
        h1 = 2 * s/a;
        h2 = 2 * s/b;
        h3 = 2 * s/c;
        printf ("\nh1 = %f\nh2 = %f\nh3 = %f", h1,h2,h3);
    }
    else
        printf("Такой треугольник не существует\n");
    getch();
    return 0;
}
```

3.4.5. Результат

```
#include "stdafx.h"
#include <stdio.h>
#include <conio.h>
#include <locale>
#include <math.h>
int main()
{
    setlocale(0, "rus");
    float a, b, c, s, p, h1, h2, h3;
    printf("Введите значение сторон треугольника\n");
    scanf_s("%f%f%f", &a, &b, &c);
    if(a>0 && b>0 && c>0 && a+b>c && a+c>b && b+c>a)
    {
        p = (a + b + c)/2;
        s = sqrt(p * (p - a) * (p - b) * (p - c));
        h1 = 2 * s/a;
        h2 = 2 * s/b;
        h3 = 2 * s/c;
        printf ("\nh1 = %f\nh2 = %f\nh3 = %f", h1,h2,h3);
    }
    else
        printf("Такой треугольник не существует\n");
    getch();
    return 0;
}
```

```
Введите значение сторон треугольника
43 23 32

h1 = 16,766651
h2 = 31,346348
h3 = 22,530188
```

3.4.6. Тесты

a	b	c	Ручной подсчёт ha	Ручной подсчёт hb	Ручной подсчёт	Програ ммный подсчёт ha	Програ ммный подсчёт hb	Програ ммный подсчёт hc
4 3	2 3	3 2	16,7666 51	31,3463 48	22,5301 88	16,7666 51	31,3463 48	22,5301 88
1 5	1 5	1 5	12,9903 81	12,9903 81	12,9903 81	12,9903 81	12,9903 81	12,9903 81
3 4	2 5	1 5	10,0696 54	13,6947 29	22,8245 49	10,0696 54	13,6947 29	22,8245 49
2 6	3 7	4 6	36,9748 00	25,9822 92	20,8988 00	36,9748 00	25,9822 92	20,8988 00

4. Номер четвертый

4.1. Постановка задачи

Вычислите площадь поверхности и объём усечённого конуса, используя функцию возведения в квадрат POW.

$$s = \pi * (R + r) * l + \pi * R^2 + \pi * r^2,$$

$$v = (1/3) * (R^2 + r^2 + R * r) * h * \pi.$$

4.2. Структурные требования

r = не должно равняться нулю

h = не должно равняться нулю

$R > r$

$l > h$

4.3. Описание работы

4.4.1. Используемые переменные

h – высота

r – радиус верху

R – радиус основания

s – Площадь поверхности усеченного конуса

v – объём усеченного конуса

4.4.2. Псевдокод программы

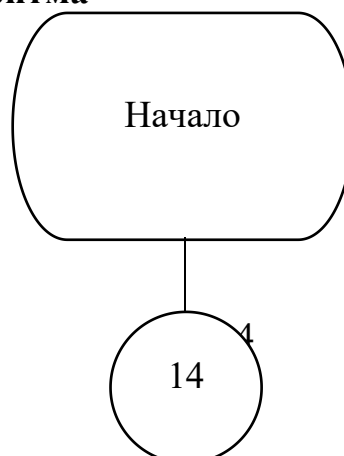
Вводим с клавиатуры r , R , k , h

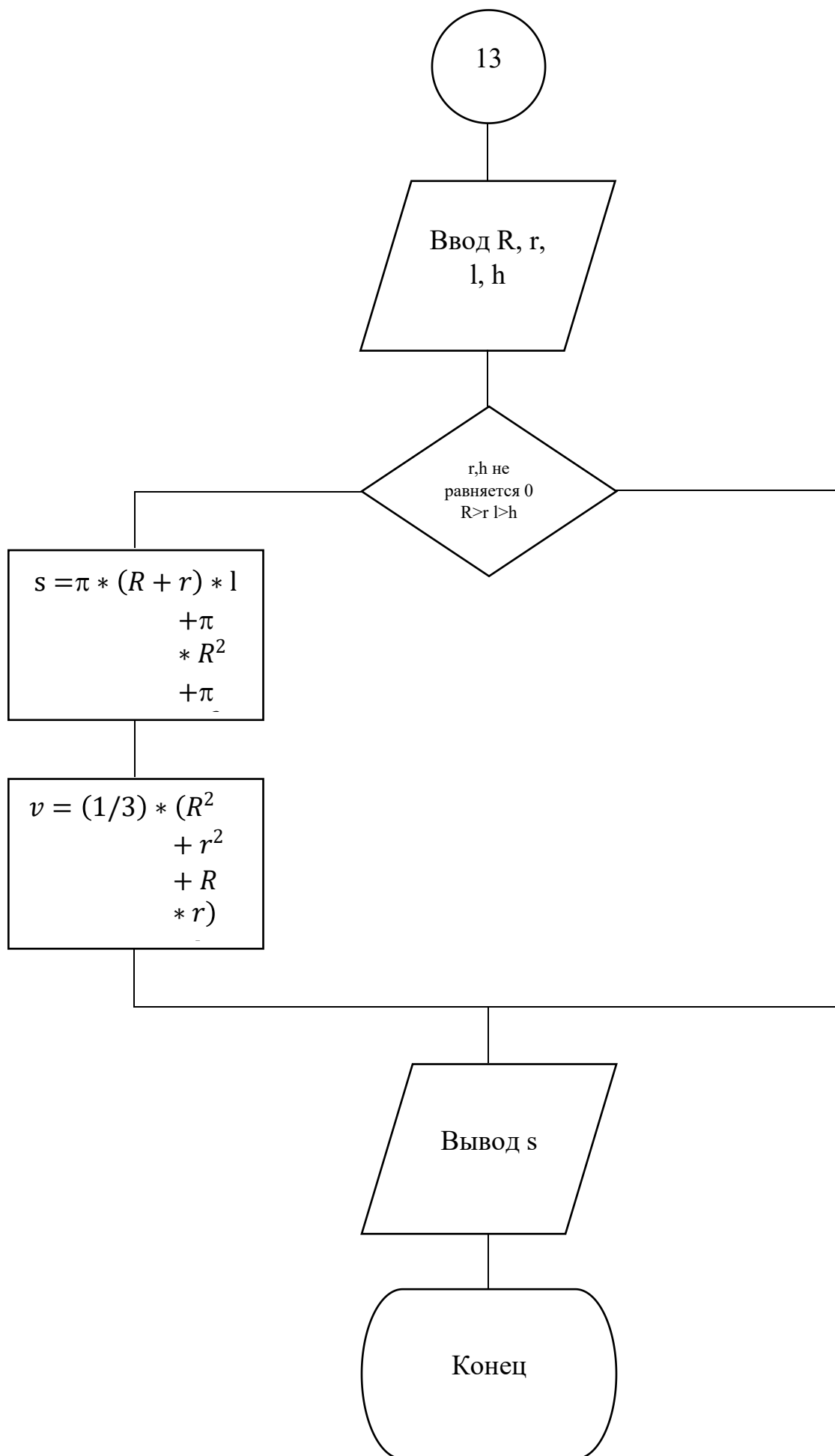
Проверяем что бы r и $h > 0$, $R > r$, $l > h$.

Выполняем $s = \pi * (R + r) * l + \pi * R^2 + \pi * r^2$,

Выполняем $v = (1/3) * (R^2 + r^2 + R * r) * h * \pi$.

4.4.3. Блок-схема алгоритма





4.4.4. Листинг программной реализации

```
#include "stdafx.h"
#include <stdio.h>
#include <conio.h>
#include <locale>
#include <math.h>
int main()
{
    setlocale(0, "rus");
    float v, s, R, r, l, h;
    printf("Введите значение переменных R, r, l и h");
    scanf_s("%f%f%f%f", &R, &r, &l, &h);
    if (r>0 && h>0 && R>r && l>h)
    {
        s = 3.14159 * (R + r) * l + 3.14159 * pow(R, 2) + 3.14159 * pow(r, 2);
        v = h * 3.14159 * (pow(R,2) + pow(r,2) + R * r) * 1/3;
        printf("\ns = %f", s);
        printf("\nv = %f", v);
    }
    else
        printf("Такой конус не существует\n");
    getch();
    return 0;
}
```

4.4.5. Результат

```
#include "stdafx.h"
#include <stdio.h>
#include <conio.h>
#include <locale>
#include <math.h>
int main()
{
    setlocale(0, "rus");
    float v, s, R, r, l, h;
    printf("Введите значение переменных R, r, l и h");
    scanf_s("%f%f%f%f", &R, &r, &l, &h);
    if (r>0 && h>0 && R>r && l>h)
    {
        s = 3.14159 * (R + r) * l + 3.14159 * pow(R, 2) + 3.14159 * pow(r, 2);
        v = h * 3.14159 * (pow(R,2) + pow(r,2) + R * r) * 1/3;
        printf("\ns = %f", s);
        printf("\nv = %f", v);
    }
    else
        printf("Такой конус не существует\n");
    getch();
    return 0;
}
```

```
Введите значение переменных R, r, l и h
24 19 9 3

s = 4159,465332
v = 4376,234863
```

4.4.6. Тесты

h	R	r	l	Ручной подсчёт s	Ручной подсчёт v	Программны й подсчёт s	Программный Подсчёт v
1 5	1 2	7	5	1753,007202	2874,554932	1753,007202	2874,554932
2 4	1 9	9	3	4159,465332	4376,234863	4159,465332	4376,234863
9 9	9 7	6 5	5 6	100373,79687 5	1689681,12500 0	100373,79687 5	1689681,12500 0

5. Номер пятый

5.1. Постановка задачи

Вычислите значение функции, используя функцию возведения e в заданную степень EXP и функцию SIN.

$$y = a * e^{-a*\omega*x} * \sin(\omega * x),$$
$$x = (\pi/2 - \varphi).$$

5.2. Структурные требования

$\varphi > 0$

$\varphi < 1$

$\omega > 0$

$\omega < 1$

5.3. Описание работы

5.5.1. Используемые переменные

φ – φ вводим с клавиатуры

ω – ω вводим с клавиатуры

x – находим по формуле $x = (\pi/2 - \varphi)$

y – находим по формуле $y = a * e^{-a*\omega*x} * \sin(\omega * x)$

5.5.2. Псевдокод программы

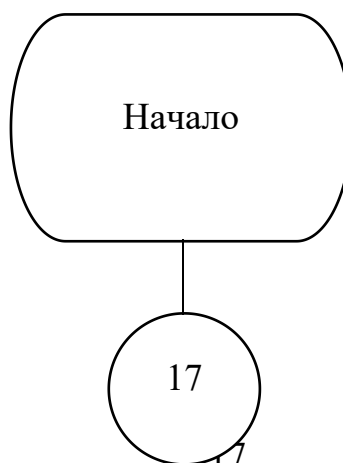
Вводим с клавиатуры φ и ω

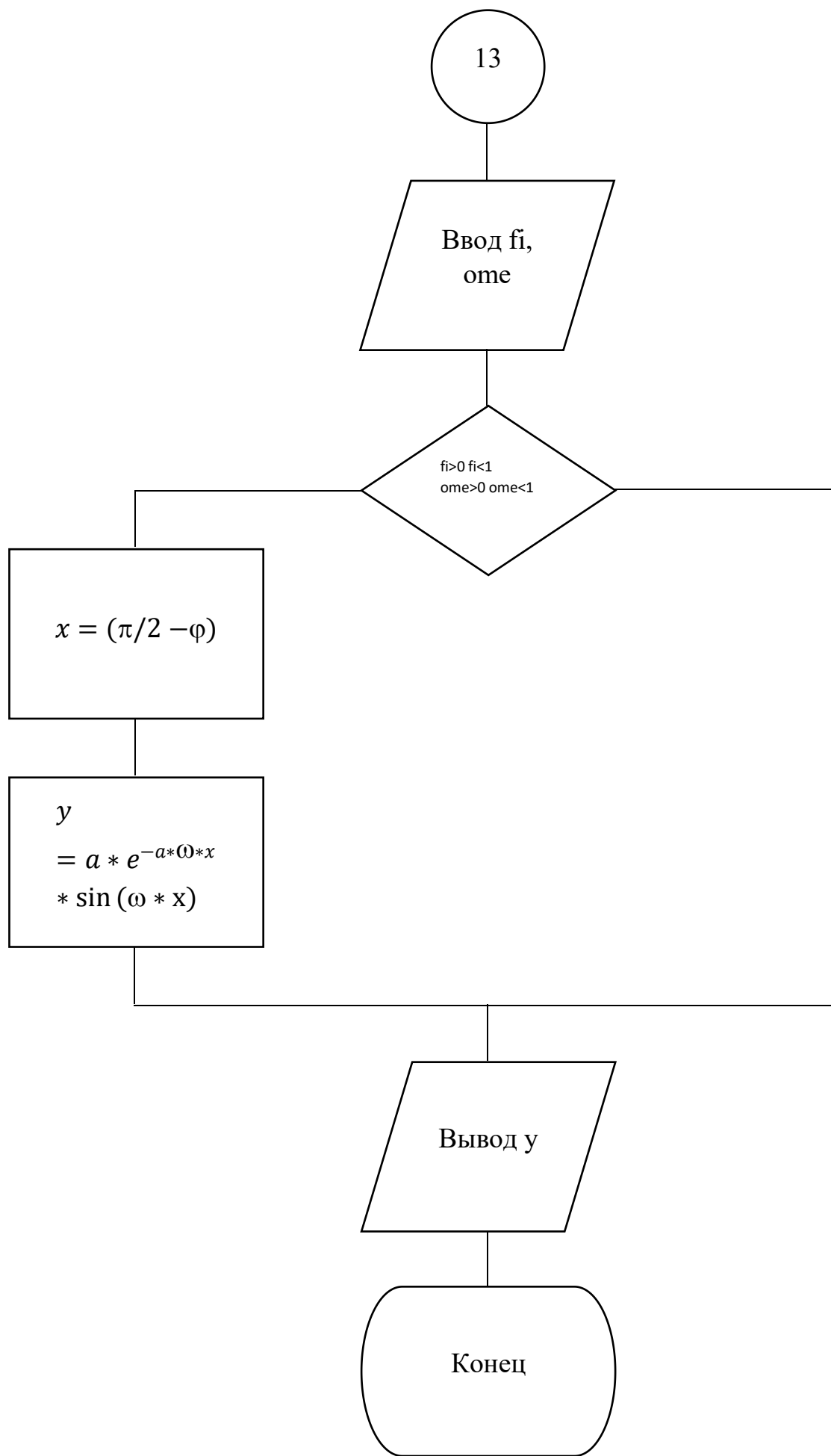
x – находим по формуле $x = (\pi/2 - \varphi)$

y – находим по формуле $y = a * e^{-a*\omega*x} * \sin(\omega * x)$

выводим y

5.5.3. Блок-схема алгоритма

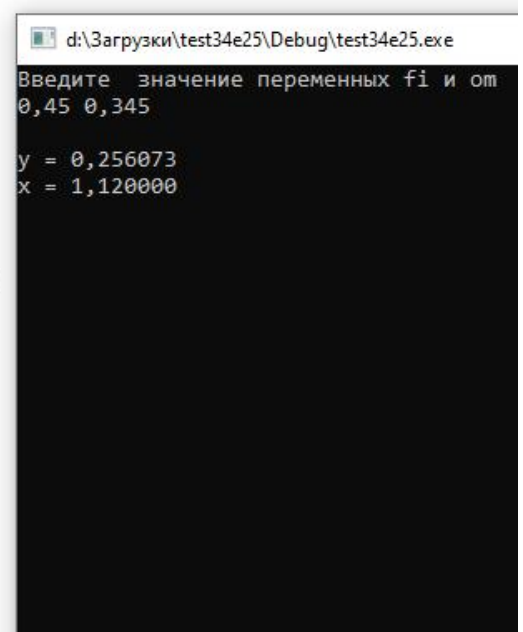




5.5.4. Листинг программной реализации

5.5.5. Результат

```
#include "stdafx.h"
#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <math.h>
int main()
{
    setlocale(0, "rus");
    float y, x, a = 1, fi, om;
    printf("Введите значение переменных fi и om");
    scanf_s("%f%f", &fi, &om);
    if (fi>0 && fi<1 && om>0 && om<1)
    {
        x = 3.14/2 - fi;
        y = a * exp(-a * om * x) * sin(om * x);
        printf("\ny = %f\nx = %f\n", y, x);
    }
    else
    {
        printf("Вы ввели недопустимые значения");
        getch();
        return 0;
    }
}
```



5.5.6. Тесты

fi	ome	Ручной подсчёт y	Программный подсчёт y	Ручной подсчёт x	Программный подсчёт x
0,45	0,345	0,256073	0,256073	1,120000	1,120000
0,1	0,1	0,126448	0,126448	1,470000	1,470000
0,14	0,15	0,171765	0,171765	1,430000	1,430000
0,5665	0,2345	0,184266	0,184266	1,003500	1,003500

6. СПИСОК ЛИТЕРАТУРЫ

1. Конова Е.А., Поллак Г.А. Алгоритмы и программы. Язык С++: Учебное пособие. - 2-е изд., стер. - СПб.: Издательство "Лань", 2017. - 384 с.
2. Седжвик Роберт. Алгоритмы на С++.: Пер. с англ. - М.: ООО "И.Д. Вильямс", 2011. - 1056 с.
3. Лафоре Р. Объектно-ориентированное программирование в С++. Классика Computer Science. 4-е изд. - СПб.: Питер, 2015. - 928 с.
4. Орлов С.А. Теория и практика языков программирования: Учебник для вузов. Стандарт 3-го поколения. - СПб.: Питер, 2014. - 688 с.
5. Павловская Т.А. С/С++. Процедурное и объектно-ориентированное программирование: Учебник для вузов. Стандарт 3-го поколения. - СПб.: Питер, 2015. - 496 с.
6. Павловская Т.А. С#. Программирование на языке высокого уровня: Учебник для вузов. - СПб.: Питер, 2014. - 432 с.
7. Плаксин М.А. Тестирование и отладка программ - для профессионалов будущих и настоящих. - М.: БИНОМ. Лаборатория знаний, 2013. - 167 с.