



МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования

«Санкт-Петербургский государственный технологический институт
(технический университет)»

УГСН	09.00.00	Информатика и вычислительная техника
Уровень образования		Высшее образование – бакалавриат
Форма обучения		Очная
Факультет		Информационных технологий и управления
Кафедра		Систем автоматизированного проектирования и управления
Учебная дисциплина		Информационные технологии и программирование
Курс	I	Группа 4304

Отчёт по контрольной работе № 1

Вариант № 26

Исполнитель:
обучающийся
группы 4304

(дата, подпись)

Рыбник Всеволод Сергеевич

Проверил:

(дата, подпись)

Корниенко Иван Григорьевич

Макарук Роман Валерьевич

Федин Алексей Константинович

Санкт-Петербург
2023

СОДЕРЖАНИЕ

1 Задание №1.....	3
1.1 Цель работы.....	3
1.2 Постановка задачи.....	3
1.3 Описание хода выполнения.....	3
1.4 Блок-схема алгоритма решения задачи.....	3
1.5 Исходный код полученного программного решения.....	5
1.6 Тестирование.....	6
1.7 Выводы по заданию №1.....	7
2 Задание №2.....	8
2.1 Цель работы.....	8
2.2 Постановка задачи.....	8
2.3 Описание хода выполнения.....	8
2.4 Блок-схема алгоритма решения задачи.....	8
2.5 Исходный код полученного программного решения.....	10
2.6 Тестирование.....	12
2.7 Выводы по заданию №2.....	12
3 Задание №3.....	13
3.1 Цель работы.....	13
3.2 Постановка задачи.....	13
3.3 Описание хода выполнения.....	13
3.4 Блок-схема алгоритма решения задачи.....	13
3.5 Исходный код полученного программного решения.....	15
3.6 Тестирование.....	16
3.7 Выводы по заданию №3.....	16

1 Задание №1

1.1 Цель работы

Изучение возможностей библиотеки <math.h>.

1.2 Постановка задачи

Разработать программу для вычисления арифметического выражения с тремя неизвестными: x, y и z, и вывода полученного результата на экран.

$$f = \ln \left| (y - \sqrt{|x|}) \cdot \left(x - \frac{y}{z + \frac{x^2}{4}} \right) \right| \quad (1)$$

1.3 Описание хода выполнения

Для выполнения задачи необходимо дополнительно изучить способы ввода для значений типа «double». Чтобы правильно решить приведённое уравнение, необходимо дополнительно изучить синтаксис необходимых функций в C. Для упрощения процесса тестирования и отладки программы, было принято решение о инициализации переменных значениями, которые предполагают ввод с клавиатуры.

1.4 Блок-схема алгоритма решения задачи

На рисунке 1 представлена блок-схема алгоритма решения вышеупомянутой задачи.

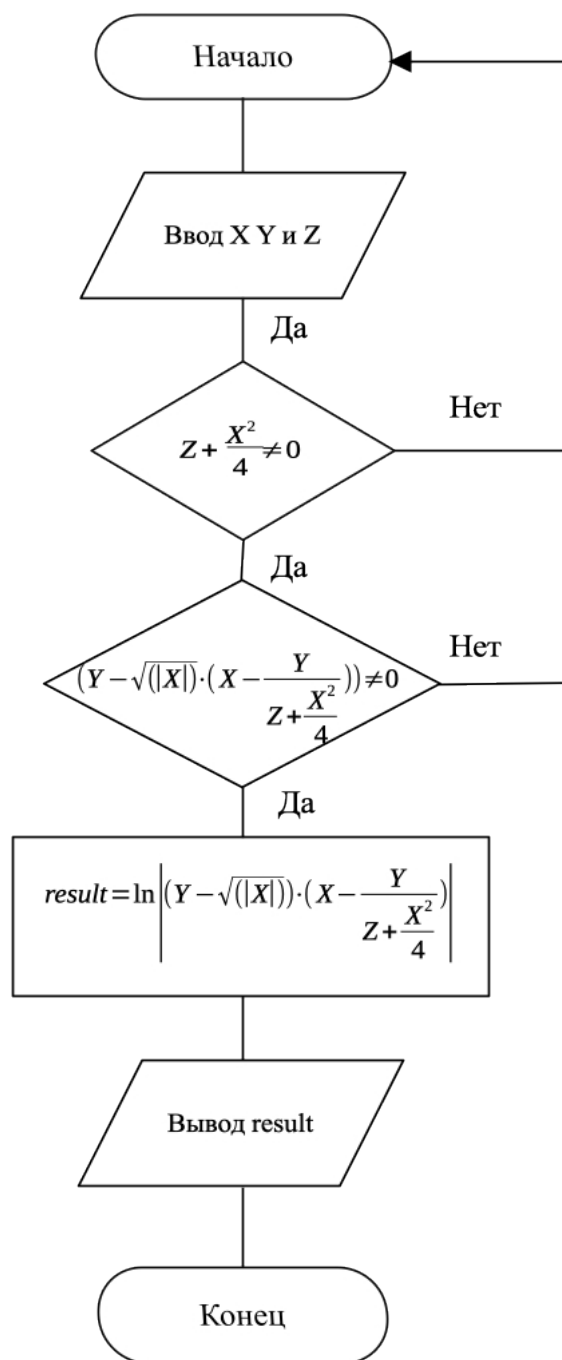


Рисунок 1 – Блок-схема алгоритма решения задачи №1

1.5 Исходный код полученного программного решения

Код файла `first_test.c`:

```
/*
*****
* Название: first_test.c *
* Задание: Первая программа в осеннем семестре *
* Автор: в.с. рыбник, СПбГТИ (ТУ), 2023 *
*****
*/

#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <time.h>
#include "first_module.h"

double GetDouble(void)
{
    char temprem, tempclear; // временный остаток
    double input = 0;
    while(true)
    {
        temprem=0;
        tempclear=0;
        if(!scanf("%lf%c",&input ,&temprem)|| temprem != '\n')
        {
            printf("Error: Invalid value for double variables.\nOne more time: ");
            while(tempclear != '\n')
                scanf("%c",&tempclear);
        }
        else
            return input;
    }
}

int main(void){
    srand(time(NULL)); // NULL - seconds - Указатель на объект типа time_t, где
    хранится кол-во секунд.
    double x = 0.0, y = 0.0, z = 0.0, result = 0;
    int loop_indicator = 1;
    int rand_manually_indicator = 1;
    char temprem = "f";
    printf("Vsevolod Rybnik Test 1 task 1 var. 26\n");
    while (loop_indicator) {
        printf("Wanna specify values yourself or get random ones? (`any num` -
        random, 0 - manually): ");
        scanf("%d", &rand_manually_indicator);
        if (rand_manually_indicator)
        {
            x = -100 + rand()%(100 + 99);
            printf("Value for x is %f\n", x);
            y = -100 + rand()%(100 + 99);
            printf("Value for y is %f\n", y);
            z = -100 + rand()%(100 + 99);
            printf("Value for z is %f\n", z);
        }
    }
}
```

```

else
{
printf("Specify values for x: ");
x = GetDouble();
printf("Specify values for y: ");
y = GetDouble();
printf("Specify values for z: ");
z = GetDouble();
}
if (!RAV(x, y, z)) {
printf("Wrong values that dont fit RAV of this expression.\n");
continue;
}
result = log_formula(x, y, z);
printf("Result: %f \n", result);
printf("Wanna new data input? (`any num` - yep, 0 - nope): ");
scanf("%d", &loop_indicator);
}
return EXIT_SUCCESS;
}

```

Код модуля first_module.h:

```

/*****
* Название: first_module.h *
* Задание: Первая программа в осеннем семестре *
* Автор: в.с. рыбник, СПбГТИ (ТУ), 2023 *
*****/

#include <stdlib.h>
#include <stdbool.h>
#include <math.h>

bool RAV(double x, double y, double z) {
return (((z + pow(x, 2)/4) != 0) && (y - sqrt(fabs(x))) * (x - y / (z +
pow(x, 2)/4)) != 0);
}

double log_formula(double x, double y, double z){
return log(fabs((y - sqrt(fabs(x))) * (x - y / (z + pow(x, 2)/4))));
}

```

1.6 Тестирование

Тестирование программы было проведено на следующих исходных данных: $x = 4$, $y = 5$, $z = 6$. Результат тестирования приведён на рисунке 2

```
Vsevolod Rybnik Test 1 task 1 var. 26
Wanna specify values yourself or get random ones? (`any num` – random, 0 – manually): 0
Specify values for x: 4
Specify values for y: 5
Specify values for z: 6
Result: 2.351375
Wanna new data input? (`any num` – yep, 0 – nope): 1
Wanna specify values yourself or get random ones? (`any num` – random, 0 – manually): 1
Value for x is 66.000000
Value for y is -6.000000
Value for z is -93.000000
Result: 6.837624
Wanna new data input? (`any num` – yep, 0 – nope): █
```

Рисунок 2 – Экранная копия результата работы разработанной программы задания №1

1.7 Выводы по заданию №1

В ходе выполнения задания №1 контрольной работы №1 были изучены основы работы с библиотекой `math.h`, изучены способы получения данных от пользователей, а также принцип наследования методов из файла в файл.

2 Задание №2

2.1 Цель работы

Изучить условные операторы в языке C, научиться их применять также разобраться в том как использовать множественные условия для одного случая.

2.2 Постановка задачи

Разработать программу для вычисления выражения и вывода полученного результата на экран. Соответствующие исходные данные пользователь вводит с клавиатуры. Выражения, а также условия, которые необходимо разработать:

$n = \begin{cases} \sin(x) + 0,5 \cdot \sqrt{ x + y \cdot z }, & y < -3 \\ 2 \cdot y \cdot \sqrt{x^2 + z}, & -3 \leq y \leq 1 \\ 3 \cdot x^3 - 2 \cdot y^2 + z, & y > 1 \end{cases}$	(2.1) (2.2) (2.3)
--	-------------------------

2.3 Описание хода выполнения

Сперва была создана блок-схема будущего алгоритма, затем изучена информация обо всех существующих условных операторах в C (if-else, switch-case), различия между ними. Для данного алгоритма было принято решение использовать if-else. Опираясь на составленную блок схему разработан алгоритм в Pelles, протестирован с несколькими данными, добавлен вывод сообщений об ошибках, которые сообщают о невозможности выполнения каких-либо методов, если переменная, в данном случае y, не удовлетворяет каким-либо условиям.

2.4 Блок-схема алгоритма решения задачи

На рисунке 3 представлена блок-схема алгоритма решения задачи №2.

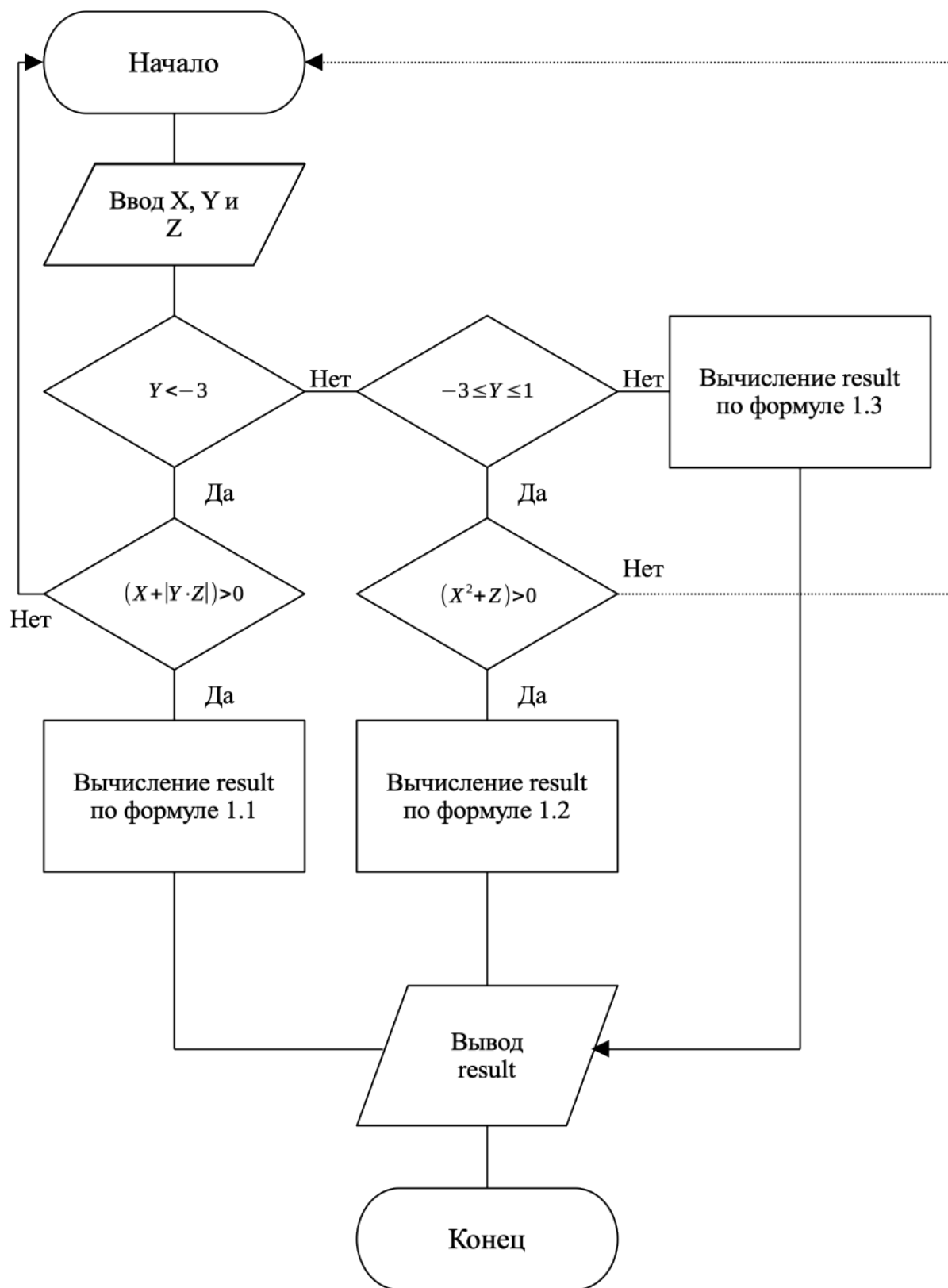


Рисунок 3 – Блок-схема алгоритма решения задачи №2

2.5 Исходный код полученного программного решения

Код файла **second_test.c**:

```

/*****
 * Название: second_test.c *
 * Задание: Вторая программа в осеннем семестре *
 * Автор: в.с. рыбник, СПбГТИ (ТУ), 2023 *
 *****/

#include <stdlib.h>
#include <stdio.h>
#include <stdbool.h>
#include <time.h>
#include "second_module.h"

double GetDouble(void)
{
    char temprem, tempclear;
    double input = 0;
    while(true)
    {
        temprem=0;
        tempclear=0;
        if(!scanf("%lf%c",&input ,&temprem)|| temprem != '\n')
        {
            printf("Error: Invalid value for double variables.\nOne more time: ");
            while(tempclear != '\n')
                scanf("%c",&tempclear);
        }
        else
            return input;
    }
}

bool RAV_for_first(double x, double y, double z)
{
    return (x + fabs(y*z)) > 0;
}

bool RAV_for_second(double x, double z)
{
    return (pow(x, 2) + z) > 0;
}

int main(void){
    srand(time(NULL));
    double x = 0.0, y = 0.0, z = 0.0, result;
    int loop_indicator = 1, rand_man_indicator = 1;
    printf("Vsevolod Rybnik Test 1 task 2 var. 26\n");
    while (loop_indicator)
    {
        result = 0.0;
        printf("Wanna specify values yourself or get random ones? (`any num` - random, 0 - manually): ");
        scanf("%d", &rand_man_indicator);
        if (!rand_man_indicator)
        {

```

```

printf("Specify values for x: ");
x = GetDouble();
printf("Specify values for y: ");
y = GetDouble();
printf("Specify values for z: ");
z = GetDouble();
}
else
{
x = -100 + rand()%(100 + 99);
printf("Value for x is %f\n", x);
y = -100 + rand()%(100 + 99);
printf("Value for y is %f\n", y);
z = -100 + rand()%(100 + 99);
printf("Value for z is %f\n", z);
}
if (y < -3)
{
if (!RAV_for_first(x, y, z))
{
printf("Wrong values that dont fit RAV of first expression.\n");
continue;
}
result = first_expression_method(x, y, z);
printf("Result: %f \n", result);
}
if (-3 <= y && y <= 1)
{
if (!RAV_for_second(x, z))
{
printf("Wrong values that dont fit RAV of second expression.\n");
continue;
}
result = second_expression_method(x, y, z);
printf("Result: %f \n", result);
}
if (y > 1)
{
result = third_expression_method(x, y, z);
printf("Result: %f \n", result);
}
printf("Wanna new data input? (`any num` - yep, 0 - nope): ");
scanf("%d", &loop_indicator);
}
return EXIT_SUCCESS;
}

```

Код модуля second_module.h:

```

/*****
* Название: second_module.h *
* Задание: Вторая программа в осеннем семестре *
* Автор: в.с. рыбник, СПбГТИ (ТУ), 2023 *
*****/

```

```

#include <math.h>

```

```

double first_expression_method(double x, double y, double z){
return sin(x) + 0.5 * sqrt(x + fabs(y*z));
}

```

```
};

double second_expression_method(double x, double y, double z){
return 2 * y * sqrt(pow(x, 2) + z);
};

double third_expression_method(double x, double y, double z){
return 3 * pow(x, 3) - 2 * pow(y, 2) + z;
};
```

2.6 Тестирование

Тестирование проводилось с 3 разными значениями, тк как условия для каждого выражения противоречивы. В-первом случае за у взято значение -1, во-втором — -5, а в-третьем — 5, x и z — 4 и 6 соответственно.

```
Vsevolod Rybnik Test 1 task 2 var. 26
Wanna specify values yourself or get random ones? (`any num` - random, 0 - manually): 1
Value for x is -74.000000
Value for y is -31.000000
Value for z is 56.000000
Result: 21.368963
Wanna new data input? (`any num` - yep, 0 - nope): 1
Wanna specify values yourself or get random ones? (`any num` - random, 0 - manually): 0
Specify values for x: 4
Specify values for y: -5
Specify values for z: 6
Result: 2.158673
Wanna new data input? (`any num` - yep, 0 - nope): 1
Wanna specify values yourself or get random ones? (`any num` - random, 0 - manually): 0
Specify values for x: 4
Specify values for y: 5
Specify values for z: 6
Result: 148.000000
Wanna new data input? (`any num` - yep, 0 - nope): 1
Wanna specify values yourself or get random ones? (`any num` - random, 0 - manually): 0
Specify values for x: 4
Specify values for y: -1
Specify values for z: 6
Result: -9.380832
Wanna new data input? (`any num` - yep, 0 - nope): █
```

Рисунок 4 – Экранные копии результата работы разработанной программы задания 2

2.7 Выводы по заданию №2

В ходе выполнения задачи я получил опыт работы с множественными условиями, узнал тонкости присвоения значений переменным в C, а также узнал о новом условном операторе switch-case в C.

3 Задание №3

3.1 Цель работы

Табулирование функции.

3.2 Постановка задачи

Разработать программу для вычисления значений функции $y=f(x)$ в интервале $[a,b]$ с шагом h и вывода полученных результатов (пары x и y) на экран.

$y = \begin{cases} \frac{\pi \cdot x}{2}, & x > 0 \\ \frac{\pi}{x} + \sin\left(\frac{\pi}{x}\right), & x < 0 \\ 0, & x = 0 \end{cases}$	(3.1)
	(3.2)
	(3.3)

3.3 Описание хода выполнения

Для выполнения данной задачи были введены две константы — $p = \pi$, $h = 0.2$.

3.4 Блок-схема алгоритма решения задачи

На рисунке 5 представлена блок-схема алгоритма решения задачи №3

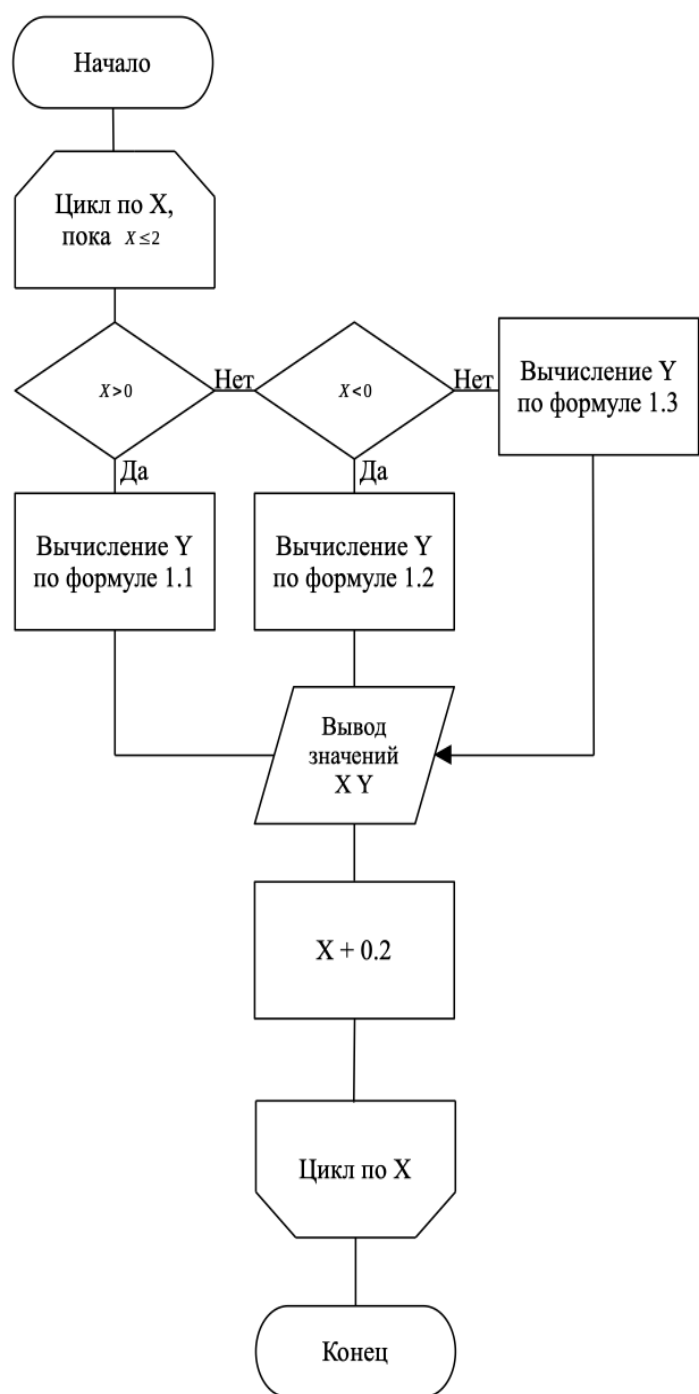


Рисунок 5 – Блок-схема алгоритма решения задачи №3

3.5 Исходный код полученного программного решения

Код файла `third_test.c`:

```
/*
 * Название: third_test.c *
 * Задание: Третья программа в осеннем семестре *
 * Автор: в.с. рыбник, СПбГТИ (ТУ), 2023 *
 */

#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <math.h>
#include "third_module.h"

int calculator(double x, double p, double h)
{
    while (x <= 2)
    {
        if (0 < x && x <= 2) {
            printf("Result: (%.3f, %.3f)\n", x, first_expression(x, p));
        }
        else if (-1 <= x && x < 0) {
            printf("Result: (%.3f, %.3f)\n", x, second_expression(x, p));
        }
        else if (x == 0)
        {
            printf("Result: (0.000, 0.000)\n");
        }
        x = round((x + h) * 1000) / 1000; // округление до 3 знаков после запятой
    }
    return EXIT_SUCCESS;
}

int main(void) {
    double x = -1;
    const double p = M_PI;
    const double h = 0.2; // учет погрешности
    int loop_indicator = 1;
    printf("Vsevolod Rybnik Test 1 task 3 var. 26\n");
    while (loop_indicator) {
        calculator(x, p, h);
        printf("Wanna keep entering values? (`any num` - yep, 0-nope) ");
        scanf("%d", &loop_indicator);
    }
    return EXIT_SUCCESS;
}
```

Код модуля `third_module.h`:

```
/*
 * Название: third_module.h *
 * Задание: Третья программа в осеннем семестре *
 * Автор: в.с. рыбник, СПбГТИ (ТУ), 2023 *
 */

#include <math.h>

double first_expression(double x, double p) {
    return (p * x) / 2;
```

```

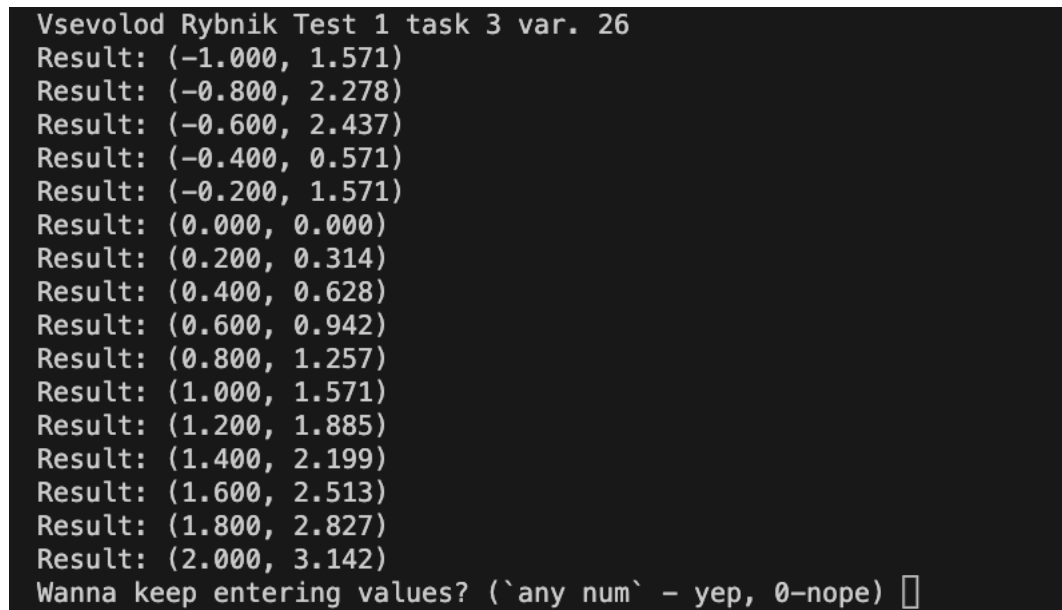
}

double second_expression(double x, double p) {
return (p/2) + sin(p/x);
}

```

3.6 Тестирование

Тестирование было проведено при $X = -1$



```

Vsevolod Rybnik Test 1 task 3 var. 26
Result: (-1.000, 1.571)
Result: (-0.800, 2.278)
Result: (-0.600, 2.437)
Result: (-0.400, 0.571)
Result: (-0.200, 1.571)
Result: (0.000, 0.000)
Result: (0.200, 0.314)
Result: (0.400, 0.628)
Result: (0.600, 0.942)
Result: (0.800, 1.257)
Result: (1.000, 1.571)
Result: (1.200, 1.885)
Result: (1.400, 2.199)
Result: (1.600, 2.513)
Result: (1.800, 2.827)
Result: (2.000, 3.142)
Wanna keep entering values? (`any num` - уер, 0-норе) █

```

Рисунок 6 – Экранная копия результата работы разработанной программы задания №1.3

3.7 Выводы по заданию №3

Были изучены и использованы принципиально новые способы получения валидации данных. Достигнута полная функциональность программы в соответствии с требованиями условия.