**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра інтелектуальних технологій**

Лабораторна робота №2  
*(вид роботи: лабораторна робота, індивідуальне завдання, курсова робота тощо)*

з дисципліни «Алгоритмізація та програмування»Тема роботи: «Розгалужений обчислювальний процес»  
**Варіант № 5**

Виконав(-ла) студент(-ка)  
групи АнД - 11  
Яковкін Микола Андрійович

Перевірив(-ла):  
ПІП викладача

Київ – 2021

**Завдання 1**

**Обчислити вираз:**

1.1 Математична постановка задачі (МПЗ).

*Вхідні дані*:

**х** – числове значення дійсного типу.

*Вихідні дані:*

**у** – результат розрахунку дійсного типу.

*Математична модель задачі:*

Якщо , то , інакше

Подання МПЗ у вигляді таблиці:

*Таблиця 1.1*

|  |  |  |
| --- | --- | --- |
| **Вхідні дані** | **Дії** | **Вихідні дані** |
| **х** – числове значення дійсного типу. | Якщо виконується , тоді ,  інакше | **у** – дійсне. |

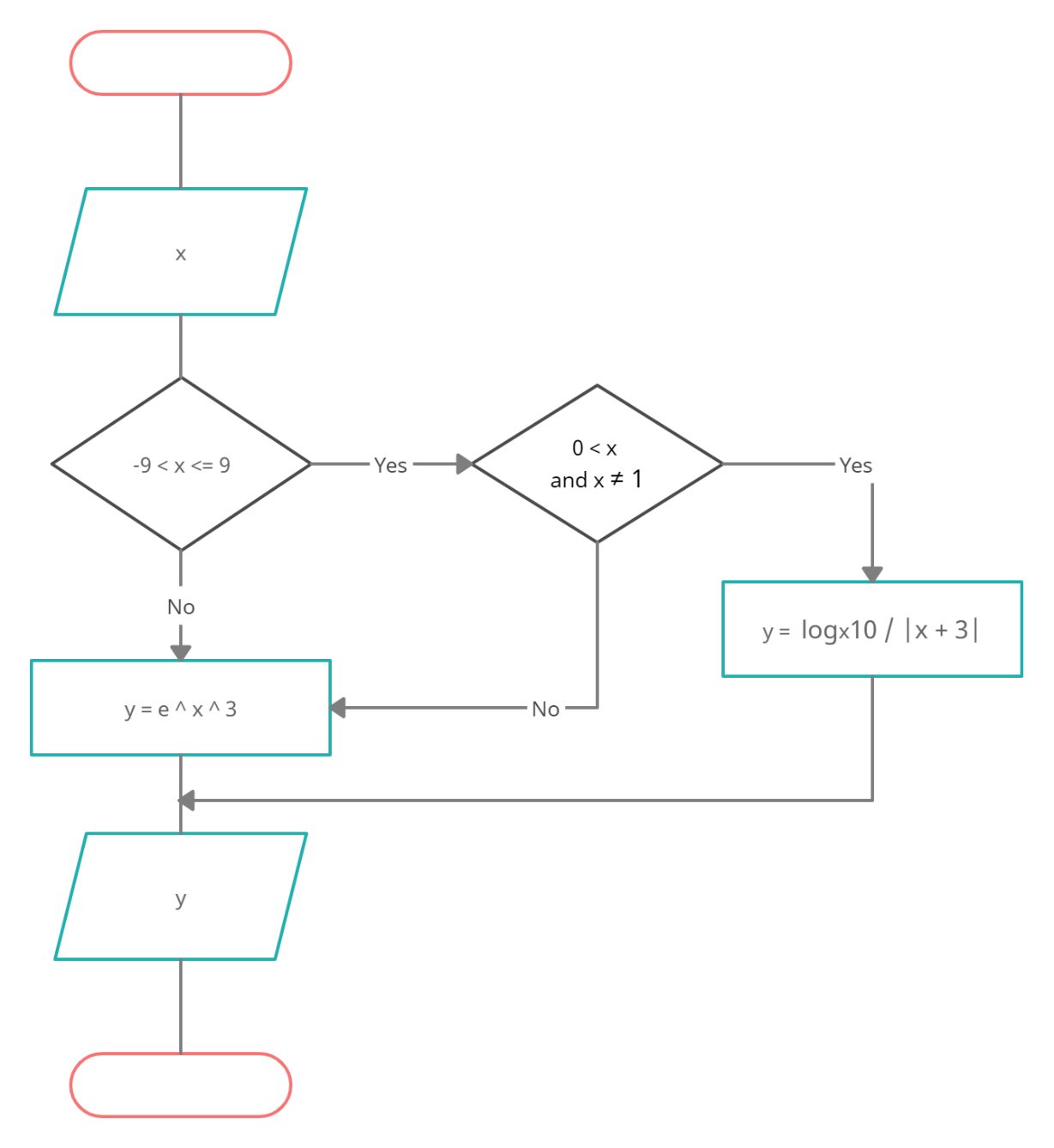


Рисунок 1 – Схема алгоритму задачі 1.

1.2 Тестовий приклад.

Для перевірки правильності МПЗ та складеної програми задамо 0 < x <= 9   
та х ≠ 1 розглянемо таки випадки:

Умова **0 < x ≤ 9** та **х ≠ 1** виконується.

* + - x = 2 =>   
      y = 0.474561

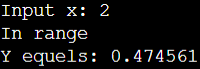


Рисунок 2 – тестування програми.

* + - x = 3 =>   
      y = 0.174659

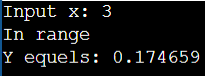
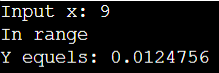


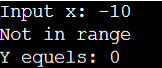
Рисунок 3 – Тестування програми

* + - x = 9 =>   
      y = 0.0124756

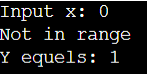


Умова **0 < x ≤ 9** та **х ≠ 1** **НЕ** виконується.

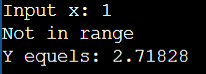
* + - x = -10 =>   
      y = 0



* + - x = 0 =>   
      y = 1



* + - x = 1 =>   
      y = 2.71828

**

*Висновок:* результати роботи програми підтверджено тестовими розрахунками.

1.3 Текст програми.

#include <iostream>

#include <cmath>

using namespace std;

int main(){

int x;

float y;

cout << "Input x: ";

cin >> x;

if(-9 < x && x <= 9 && 0 < x && x != 1){

y = (log(10) / log(x)) / fabs(x \* x + 3);

cout << "In range";

}

else{

y = exp(x\*x\*x);

cout << "Not in range" << endl;

cout << "Y equels: " << y << endl;

return 0;

}

**Завдання 2**

**Вивести на екран повідомлення, якщо точка з координатними (х, у) потрапляє у задану область.**

2.1 Математична постановка задачі (МПЗ).

*Вхідні дані:*

y – координата точки.

х – координата точки.

*Вихідні дані:*

**Т** – текстове повідомлення

*Математична модель задачі:*

Якщо x Є [-4; 4] та y Є [-4 ; 4] та (x - 2) \* (x - 2) + y \* y > 4то точка належить проміжку, інакше – ні.

Подання МПЗ у вигляді таблиці:

Таблиця 2.1:

|  |  |  |
| --- | --- | --- |
| **Вхідні дані** | **Дії** | **Вихідні дані** |
| **х** – х - координата точки;  **у** – у - координата точки. | Якщо виконується  [-4; 4] та y Є [-4 ; 4] та  (x - 2) \* (x - 2) + y \* y > 4  тоді точка належить,  інакше – ні | **Т** – текстове повідомлення |

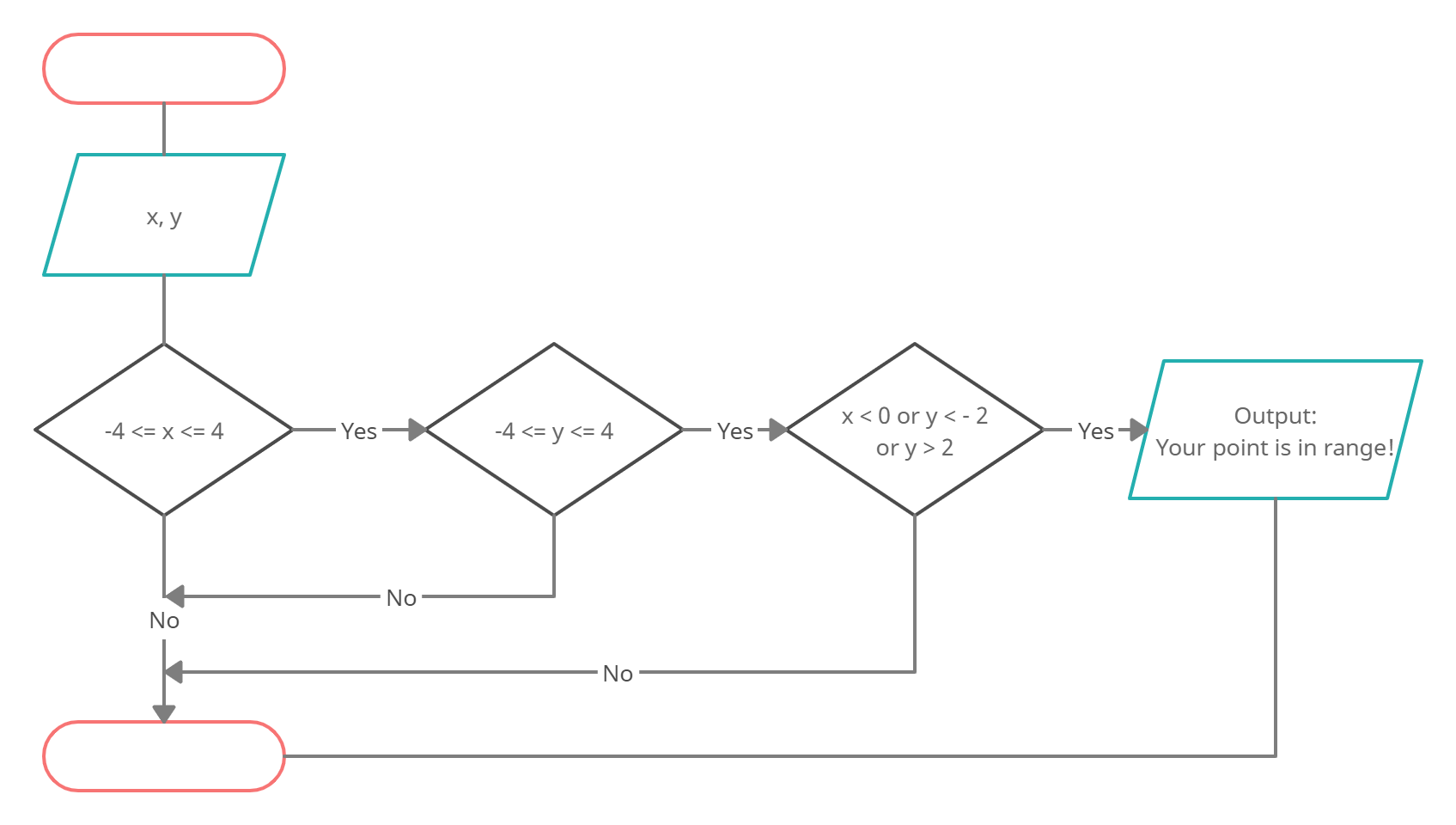


Рисунок 2.1 – Схема алгоритму задачі 2.

2.2 Тестовий приклад.

Для перевірки правильності МПЗ та складеної програми задамо -4 ≤ x < 0  
та -4 ≤ y < -2 або 2 < y ≤ 4 розглянемо таки випадки:

Умова -**4 ≤ x < 0** та (**x - 2) \* (x - 2) + y \* y > 4** виконується.

* + - x = -1, y = 3   
      Координати задовільнять умови, тому точка входить у область.

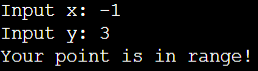


Рисунок 2.2 – тестування програми

* + - x = - 2, y = -2   
      Координати задовільнять умови, тому точка входить у область.

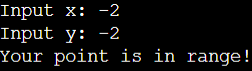


Рисунок 2.3 – Тестування програми

* + - x = - 4, y = 4  
      Координати задовільнять умови, тому точка входить у область.

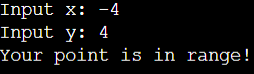


Рисунок 2.4 – Тестування програми

Умова -**4 ≤ x < 0** або (**x - 2) \* (x - 2) + y \* y > 4** *НЕ* виконується.

* х = 2, у = 1  
  Координати х та у не входять у ОДЗ, тож точка не входить у область.

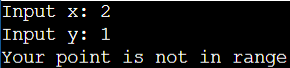


Рисунок 2.5 – Тестування програми

* x = 0, y = 0  
  Координата х та у не входять у ОДЗ, тож точка не входить у область.

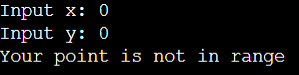


Рисунок 2.6 – Тестування програми

* х = 5, y = 4  
  Координата х не входить у ОДЗ, тож точка не входить у область.

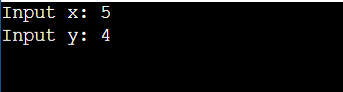


Рисунок 2.7 – Тестування програми

*Висновок:* результати роботи програми підтверджено тестовими розрахунками.

Текст програми.

#include <iostream>

#include <cmath>

using namespace std;

int main(){

int x, y;

cout << "Input x: ";

cin >> x;

cout << "Input y: ";

cin >> y;

if(-4 <= x && x <= 4 && -4 <= y && y <= 4){

if((x - 2) \* (x - 2) + y \* y > 4 ){

cout << "Your point is in range!";

}

else{

cout << "Your point is not in range";

}

}

return 0;

}

**Задані три змінні цілого типу А, В, С. Якщо хоча б одна**

**з них парна, то замінити кожну змінну на середнє значення двох інших, інакше**

**кожну змінну зменшити вдвічі.**

Математична постановка задачі (МПЗ).

*Вхідні дані:*

**А** – числове значення дійсного типу;

**В** - числове значення дійсного типу;

**С** - числове значення дійсного типу.

*Вихідні дані:*

**А** –дійсного типу;

**В** – дійсного типу;

**С** – дійсного типу.

*Математична модель задачі:*

Якщо {А / 2} = 0, або {B / 2} = 0, або {C / 2} = 0, то А = [(В + С) / 2],

В = [(А + С) / 2] та С = [(В + А) / 2], інакше С = [C / 2], В = [B / 2] та А = [A / 2].

Подання МПЗ у вигляді таблиці:

Таблиця 3

|  |  |  |
| --- | --- | --- |
| Вхідні дані | Дії | Вихідні дані |
| **А**, **B** і **С** – числові значення дійсного типу | Якщо виконується: {А / 2} = 0, або {B / 2} = 0, або {C / 2} = 0  тоді А = [(В + С) / 2],  В = [(А + С) / 2] та   С = [(В + А) / 2]  інакше С = [C / 2],  В = [B / 2] та А = [A / 2] | **А**, **В** і **С** – дійсного типу. |

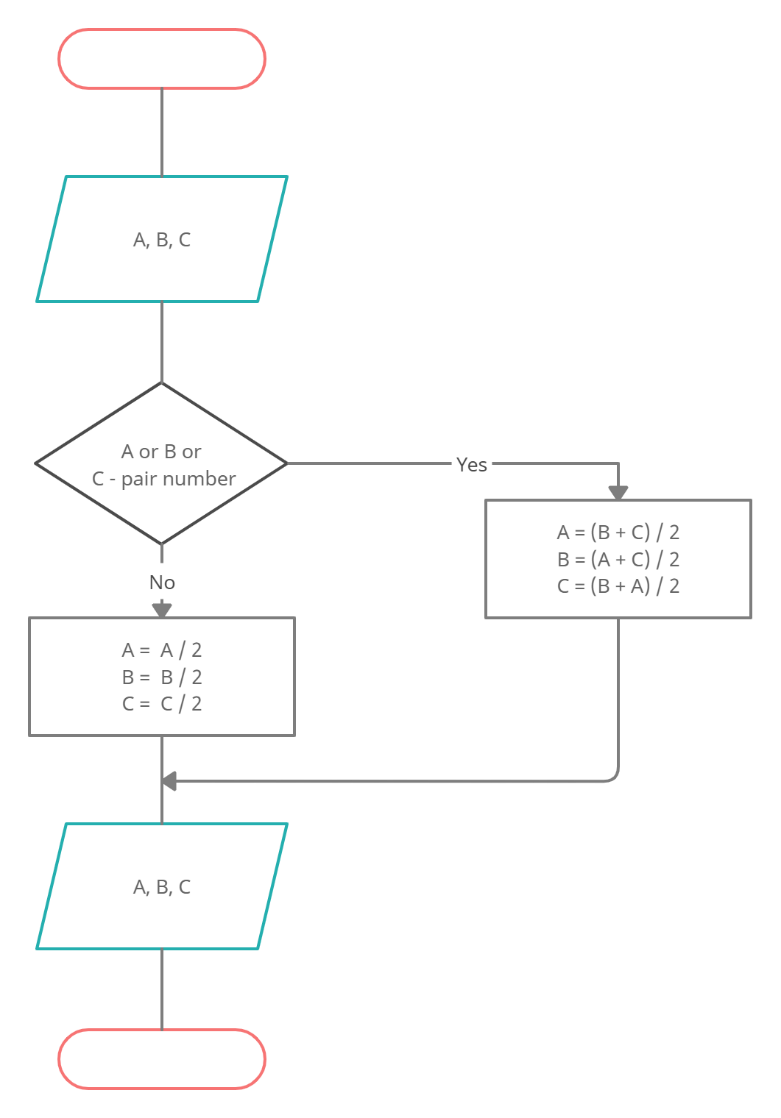


Рисунок 3 – Схема алгоритму задачі 3.

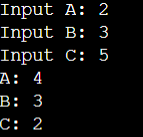
Тестовий приклад

Для перевірки правильності МПЗ та складеної програми задамо що або {А / 2} = 0, або {B / 2} = 0 чи {C / 2} = 0 розглянемо таки випадки:

Умова {**А / 2} = 0**, або {**B / 2} = 0** чи **{C / 2} = 0** виконується.

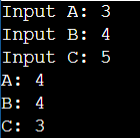
* А = 2, В = 3, С = 5. А – парне число тому умова задовольняється:

А = [(3 + 5) / 2], B = [(2 + 5) / 2] і С = [(3 + 2) / 2]  
А = 4  
В = 3  
С = 2



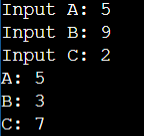
* А = 3, В = 4, С = 5. В – парне число тому умова задовольняється:

А = [(4 + 5) / 2], B = [(3 + 5) / 2] і С = [(4 + 3) / 2]  
А = 4  
В = 4  
С = 3



* А = 5, В = 9, С = 2. С – парне число тому умова задовольняється:

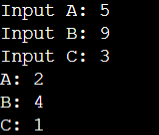
А = [(9 + 2) / 2], B = [(5 + 2) / 2] і С = [(9 + 5) / 2]  
А = 5  
В = 3  
С = 7



Умова **{А / 2} = 0**, або **{B / 2 = 0}** чи **{C / 2 = 0} НЕ** виконується.

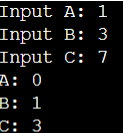
* А = 5, В = 9, С = 3. А, В і С – непарні числа тому умова не задовольняється.

А = [5 / 2], B = [9 / 2] і С = [3 / 2]  
А = 2  
В = 4  
С = 1



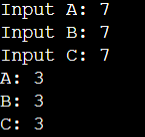
* А = 1, В = 3, С = 7. А, В і С – непарні числа тому умова не задовольняється.

А = [1 / 2], B = [3 / 2] і С = [7 / 2]  
А = 0  
В = 1  
С = 3



* А = 7, В = 7, С = 7. А, В і С – непарні числа тому умова не задовольняється.

А = [7 / 2], B = [7 / 2] і С = [7 / 2]  
А = 3  
В = 3  
С = 3



*Висновок:* результати роботи програми підтверджено тестовими розрахунками.

Текст програми.

#include <iostream>

using namespace std;

int main(){

int A, B, C, N, K;

cout << "Input A: ";

cin >> A;

cout << "Input B: ";

cin >> B;

cout << "Input C: ";

cin >> C;

if(!(A % 2) || !(B % 2) || !(C % 2) ){

N = A;

K = B;

A = (B + C) / 2;

B = (N + C) / 2;

C = (K + N) / 2;

}

else{

A /= 2;

B /= 2;

C /= 2;

}

cout << "A: " << A << endl;

cout << "B: " << B << endl;

cout << "C: " << C << endl;

return 0;

}

**Вивести значення інших елементів даного трикутника**

Математична постановка задачі (МПЗ).

Вхідні дані:  
**argument** - номер елемента. Числове значення дійсного типу.   
**х** – числове значення дійсного типу елемента.

Вихідні дані:

**а** – значення сторони дійсного типу.  
**r** – значення радіуса вписаного кола дійсного типу.  
**R** – значення радіуса описаного кола дійсного типу.  
**S** – значення площі трикутника дійсного типу.

Математична модель задачі:

(**argument**):

1, а = х, r = a / (2 \* ), R = a / , S = .  
2, r = x, a = r \* 2 \* , R = a / , S = .

3, R = x, a = R \* , r = a / (2 \* ), S = .  
4, S = x, a = , r = a / (2 \* ), R = a / .

Подання МПЗ у вигляді таблиці:

Таблиця 4

|  |  |  |
| --- | --- | --- |
| **Вхідні дані** | **Дії** | **Вихідні дані** |
| **argument** - номер елемента. Числове значення дійсного типу.  **х** – числове значення дійсного типу елемента. | 1, а = х, r = a / (2 \* ), R = a / , S = . 2, r = x, a = r \* 2 \* , R = a / , S = .  3, R = x, a = R \* , r = a / (2 \* ), S = . 4, S = x, a = , r = a / (2 \* ), R = a / .  5, “Ой... Щось трапилось.” | **a**, **r**, **R**, **S** – дійсного типу. |

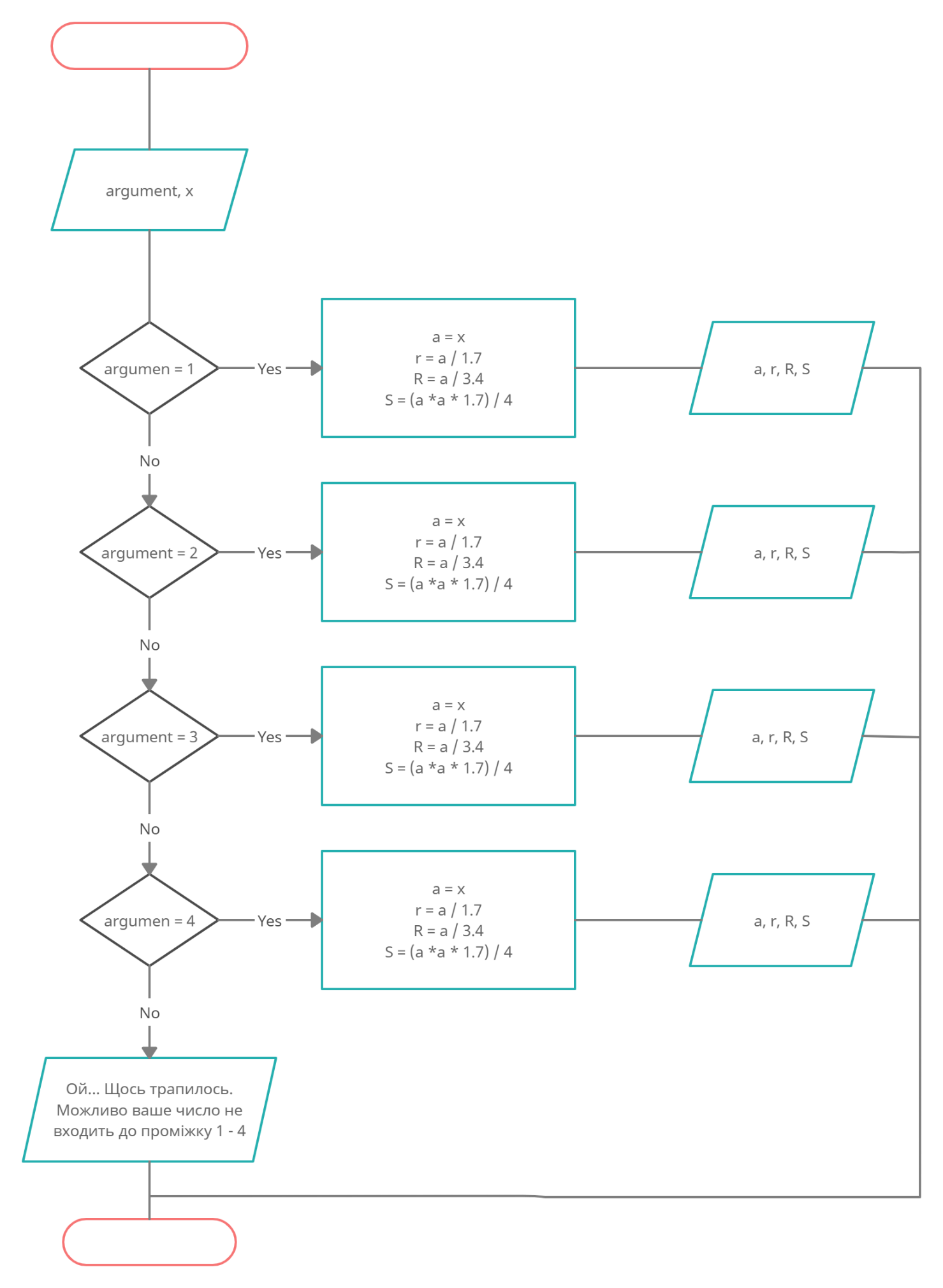


Рисунок 4 – Схема алгоритму задачі 4.

Тестовий приклад та аналіз результатів роботи програми.

|  |  |  |
| --- | --- | --- |
| **Вхідні дані** | **Очікуваний результат** | **Результат роботи програми** |
| Argument = 1 x = 4 | a = 4, r = 1.1547,  R = 2.3094, S = 6.9282. |  |
| Argument = 2 x = 3 | a = 10.3923, r = 3, R = 6, S = 46.7654. |  |
| Argument = 3 x = 2 | a = 3.4641, r =1, R = 2, S = 5.19615. |  |
| Argument = 4 x = 1 | a = 1.51967, r = 0.438691,  R = 0.877383, S = 1. |  |
| Argument = 5 x = 3 | Ой... Щось трапилось. Можливо ваше число не входить до проміжку 1 - 4 |  |

*Висновок:* результати роботи програми підтверджено тестовими розрахунками.

Текст програми.

#include <iostream>

#include <cmath>

#include <locale>

using namespace std;

int main(){

setlocale(LC\_ALL, "Ukranian");

int argument;

float a, S, r, R, x ;

cout << "Напишіть число від 1 до 4: ";

cin >> argument;

cout << "Введіть його значення: ";

cin >> x;

switch(argument)

{

case 1:

{

a = x;

R = a / sqrt(3);

r = a / (2 \* sqrt(3));

S = a \* a \* sqrt(3) / 4;

break;

}

case 2:

{

r = x;

a = r \* 2 \* sqrt(3);

R = a / sqrt(3);

S = a \* a \* sqrt(3) / 4;

break;

}

case 3:

{

R = x;

a = R \* sqrt(3);

r = a / (2 \* sqrt(3));

S = a \* a \* sqrt(3) / 4;

break;

}

case 4:

{

S = x;

a = sqrt(S \* 4 / sqrt(3));

R = a / sqrt(3);

r = a / (2 \* sqrt(3));

break;

}

default:

{

cout << "Ой... Щось трапилось. Можливо ваше число не входить до проміжку 1 - 4";

break;

}

}

cout << "a: " << a << endl;

cout << "r: " << r << endl;

cout << "R: " << R << endl;

cout << "S: " << S << endl;

return 0;

}

**Задане ціле К, що визначає номер кольору у веселці. Вивести на екран**

**повідомлення, чи є цей колір теплим чи холодним.**

Математична постановка задачі (МПЗ).

Вхідні дані:  
**colour** – числове значення дійсного типу, що відповідає номеру кольору у веселці.

Вихідні дані:  
Т – текстове повідомлення, яке визначає “холодний” чи “теплий” це колір.

Математична модель задачі:  
T(colour):

1, “теплий”  
2, “теплий”  
3, “теплий”  
4, “холодний”  
5, “холодний”  
6, “холодний”  
7, “холодний”  
(8 - ∞), "Ой... Щось не те. Спробуйте ще раз!"

Подання МПЗ у вигляді таблиці:

Таблиця 5

|  |  |  |
| --- | --- | --- |
| **Вхідні дані** | **Дії** | **Вихідні дані** |
| **Colour –** дійсного типу. | 1, “теплий” 2, “теплий” 3, “теплий” 4, “холодний” 5, “холодний” 6, “холодний” 7, “холодний” (8 - ∞), "Ой... Щось не те. Спробуйте ще раз!" | **Т** – текстове повідомлення |

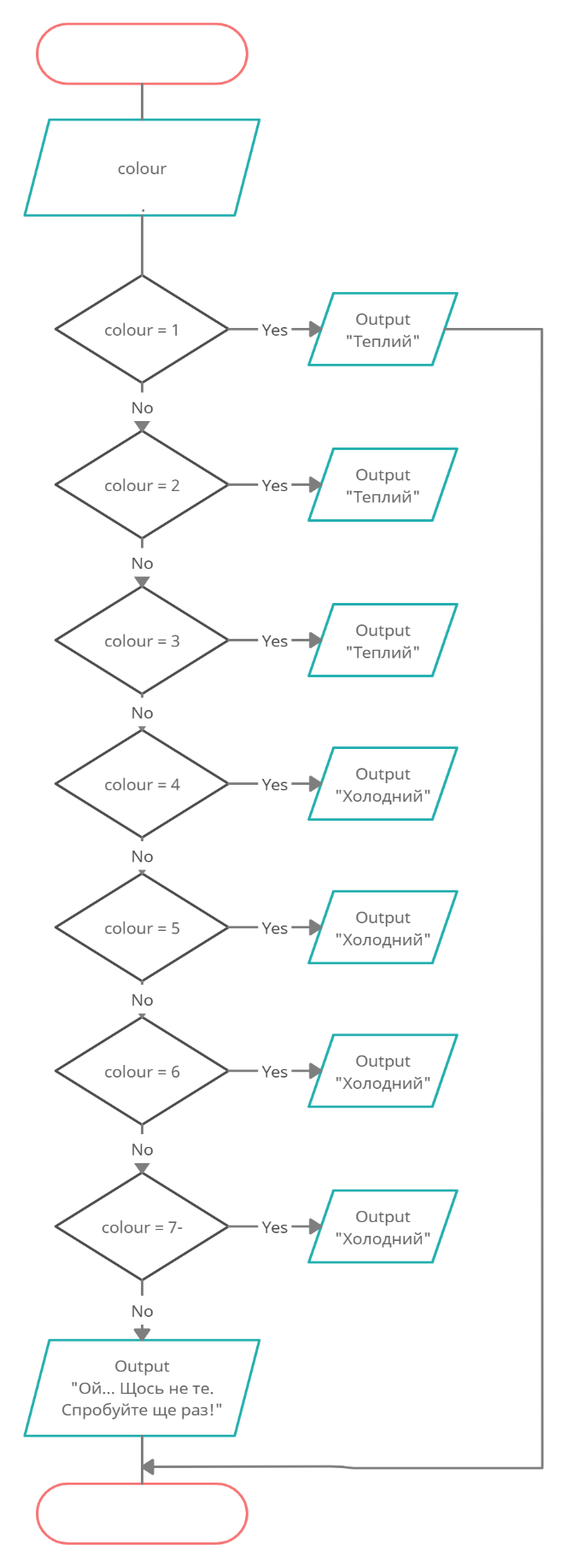


Рисунок 5 – Схема алгоритму задачі 5.

Тестовий приклад та аналіз результатів роботи програми.

|  |  |  |
| --- | --- | --- |
| **Вхідні дані** | **Очікуваний результат** | **Результат роботи програми** |
| colour = 1 | “теплий” |  |
| colour = 2 | “теплий” |  |
| colour = 3 | “теплий” |  |
| colour = 4 | “холодний” |  |
| colour = 5 | “холодний” |  |
| colour = 6 | “холодний” |  |
| colour = 7 | “холодний” |  |
| colour = 9 | "Ой... Щось не те. Спробуйте ще раз!" |  |

*Висновок:* результати роботи програми підтверджено тестовими розрахунками.

Текст програми.

#include <iostream>

#include <string>

#include <locale>

using namespace std;

int main(){

setlocale(LC\_ALL, "Ukranian");

int colour;

string introduction("Червоний: 1; \nПомаранчевий: 2; \nЖовтий: 3; \nЗелений: 4; \nГолубий: 5; \nСиній: 6; \nФіолетовий.");

cout << introduction;

cout << "Привіт! Введіть, будь ласка, значення кольору стрічки: " << endl;

cin >> colour;

switch(colour)

{

case 1: case 2: case 3: cout << "Теплий"; break;

case 4: case 5: case 6: case 7: cout << "Холодний"; break;

default:cout << "Ой... Щось не те. Спробуйте ще раз!"; break;

}

return 0;

}