**КПІ ім. Ігоря Сікорського**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

**Звіт до комп‘ютерного практикуму з курсу**

**“Основи програмування ”**

Прийняв :

Виконав:

Студент групи ІП-23

Зубарев М.К.

Асистент кафедри ІПІ

Пархоменко А. В.

08 січня 2023 р.

**Київ 2023**

**Комп‘ютерний практикум № 3**

**Тема:** *Програмування розгалужених алгоритмів.*

**Завдання 1:** Написати програму для обчислення числа із заданою точністю.

**Текст програми:**

#include <math.h>

#include <stdio.h>

#include <conio.h>

int main() {

float x, k, yk;

double e;

float y = 1;

float delta = 1;

printf("Enter X: ");

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

printf("Enter K: ");

scanf\_s("%f", &k);

printf("Enter E: ");

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

if(k == 0) {

printf("Wrong!k can't be 0\n");

return 1;

}

if (k > 0 && ((int)k % 2 == 0) && x < 0) {

printf("If k more than 0, x must be more than 0 too\n");

return 1;

}

if (k < 0 && (int)k % 2 != 0 && x == 0) {

printf("If k less than 0 and k/2 isn't equal to 0, x can't be 0\n");

return 1;

}

if (e1 >= 1 || e1 < 0 || e1 == 0) {

printf("Epsilon interval from 0 to 1\n");

return 1;

}

short int neg\_k = (k > 0) ? 0 : 1;

k = fabsf(k);

while (fabsf(delta) >= e1) {

delta = (1 / k) \* ((x / powf(y, k - 1)) - y);

y = y + delta;

if (y == 0) {

y = -1;

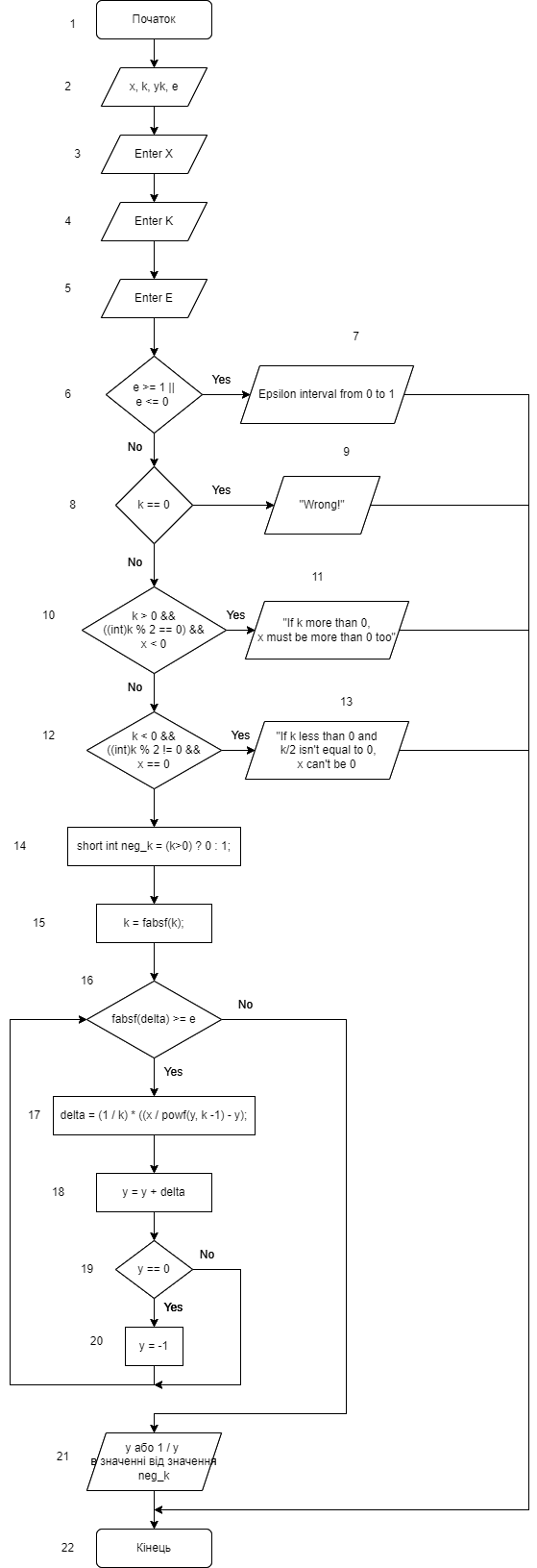
}

}

printf("y=%.\*lf\n", e, neg\_k ? 1 / y : y);

}

Блок схема до програми



Введені та одержані результати

Зображення, що містить текст

Автоматично згенерований опис1.

Зображення, що містить текст

Автоматично згенерований опис2.

Зображення, що містить текст

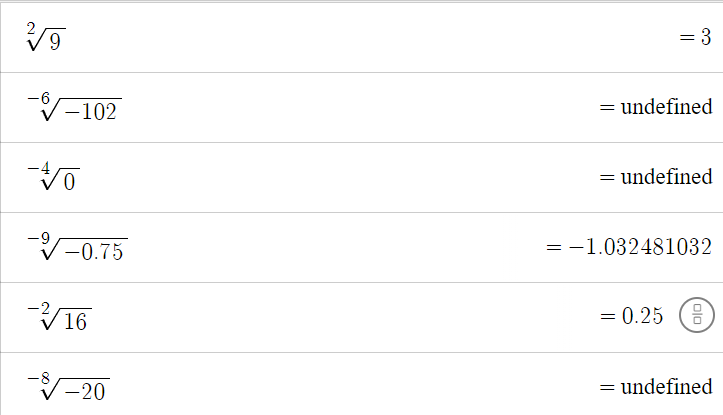
Автоматично згенерований опис3.

Зображення, що містить текст

Автоматично згенерований опис4.

Зображення, що містить текст

Автоматично згенерований опис5.



1.

2.

3.

4.

5.

6.

Висновок: Тестові приклади для перевірки коректності розрахунків відповідають отриманим. Програма працює коректно. Програма вирішує поставлене завдання.

**Завдання 2:** Написати програму – календар, яка за введеною датою виводить день тижня прописом.

**Текст програми:**

#include <stdio.h>

#include <math.h>

#include <conio.h>

int year, month, date, n, day;

char c;

int main() {

printf("Enter a date:");

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

printf("Enter a month:");

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

printf("Enter a year:");

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

int leap\_year = ((year % 100 != 0 && year % 4 == 0) || (year % 100 == 0 && year % 400 == 0));

if (month > 2) {

n = 0;

}

if (leap\_year && month <= 2) {

n = 1;

}

if (!leap\_year && month <= 2) {

n = 2;

}

if (month == 4 || month == 6 || month == 9 || month == 11 && day > 30) {

printf("\nIn this month only 30 days\n");

return 1;

}

if (month < 0 || month > 12 || date < 0 || date > 31) {

printf("\nWrite the correct value! \n");

return 1;

}

if ((!leap\_year && month == 2) && (date >= 29 && date <= 31)) {

printf("\nOnly in leap year in February 29 days \n");

return 1;

}

int day = ((int)(365.25 \* year) + (int)(30.56 \* month) + date + n) % 7;

switch (day) {

case 0:

printf("\nMonday \n");

break;

case 1:

printf("\nTuesday \n");

break;

case 2:

printf("\nWednesday \n");

break;

case 3:

printf("\nThursday \n");

break;

case 4:

printf("\nFriday \n");

break;

case 5:

printf("\nSaturday \n");

break;

case 6:

printf("\nSunday \n");

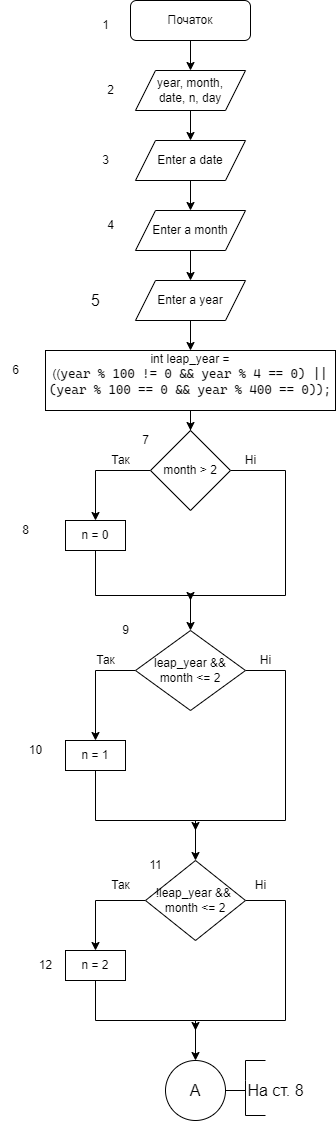
break;

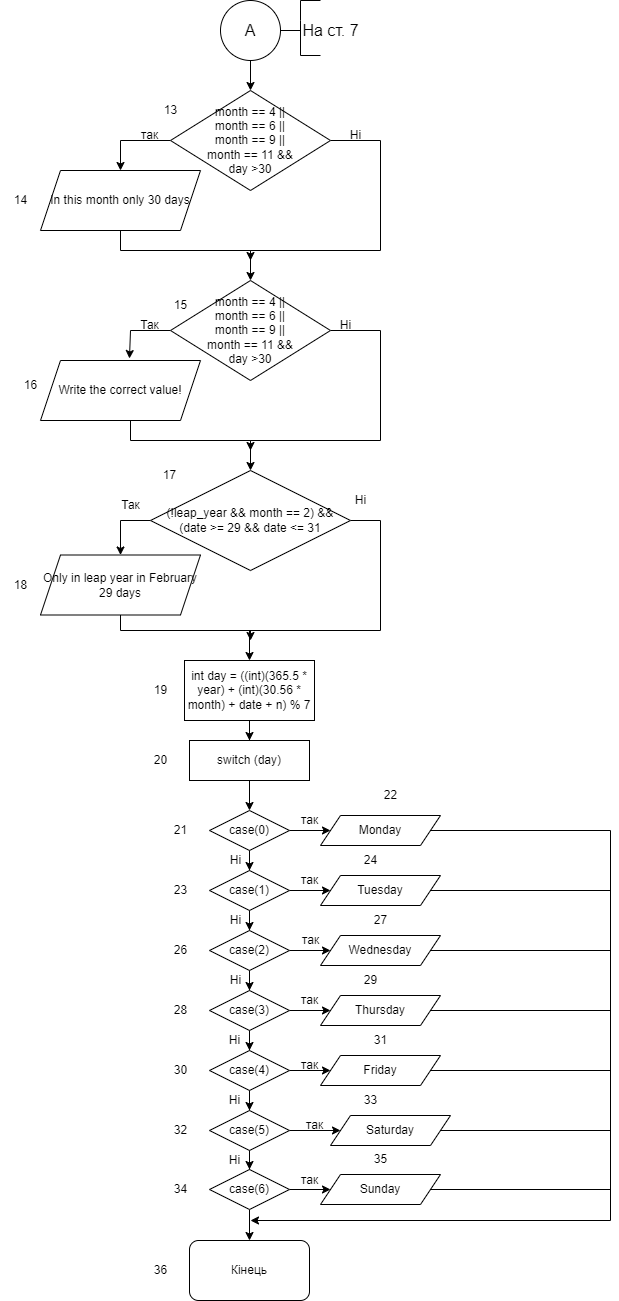
}

return 1;

}

Блок схема до програми





Введені та одержані результати

Зображення, що містить текст

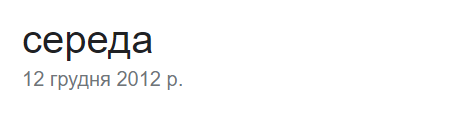
Автоматично згенерований опис1.

Зображення, що містить текст

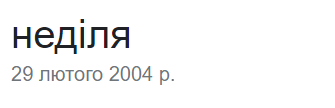
Автоматично згенерований опис2.

3. Зображення, що містить текст

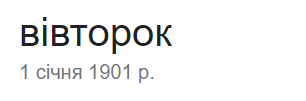
Автоматично згенерований опис



1.



2.



3.

Висновок: Тестові приклади для перевірки коректності розрахунків відповідають отриманим. Програма працює коректно. Програма вирішує поставлене завдання.

**Завдання 3:** Написати програму для розв’язання кубічного рівняння:

**Текст програми:**

#include <stdio.h>

#define \_USE\_MATH\_DEFINES

#include <math.h>

int main() {

float a, b, c, p, y, q, u, v, d, fi, r, y1, x1, x2, pol\_r, x3;

printf("\nEnter coefficient A:");

scanf\_s("%f", &a);

printf("Enter coefficient B:");

scanf\_s("%f", &b);

printf("Enter coefficient C:");

scanf\_s("%f", &c);

printf("a= %.3f b= %.3f c= %.3f\n", a, b, c);

p = b - (powf(a, 2) / 3);

q = (2 \* powf(a, 3)) / 27 - (a \* b) / 3 + c;

d = powf(p, 3) / 27 + powf(q, 2) / 4;

if (d > 0) {

y = (-q / 2) + sqrtf(d);

u = (y > 0) ? (powf(y, 1 / 3.)) : (powf(fabsf(y), 1 / 3.) \* -1);

if (u == 0) {

printf("Error");

return 1;

}

v = -p / (3 \* u);

y1 = u + v;

x2 = -(y1 / 2) - (a / 3);

pol\_r = sqrtf(3) \* (u - v) / 2;

x1 = y1 - (a / 3);

printf("\nx1 = %.4f\n", x1);

printf("x2 = %.4f + i\*%.4f\n", x2, pol\_r);

printf("x2 = %.4f - i\*%.4f\n", x2, pol\_r);

return 1;

}

else if (d < 0) {

if (powf(p, 3) / -27 >= 0) {

r = sqrtf(powf(p, 3) / -27);

fi = acosf(-q / (2 \* r));

x1 = (2 \* fabsf(cbrtf(r)) \* cosf(fi / 3)) - (a / 3);

x2 = (2 \* fabsf(cbrtf(r)) \* cosf((fi + 2 \* M\_PI) / 3)) - (a / 3);

x3 = (2 \* fabsf(cbrtf(r)) \* cosf((fi + 4 \* M\_PI) / 3)) - (a / 3);

printf("\nx1 = %.4f\n", x1);

printf("x2 = %.4f\n", x2);

printf("x3 = %.4f\n", x3);

return 1;

}

}

else {

x1 = ((3 \* q) / p) - (a / 3);

x2 = (-1 \* (3 \* q) / (2 \* p)) - (a / 3);

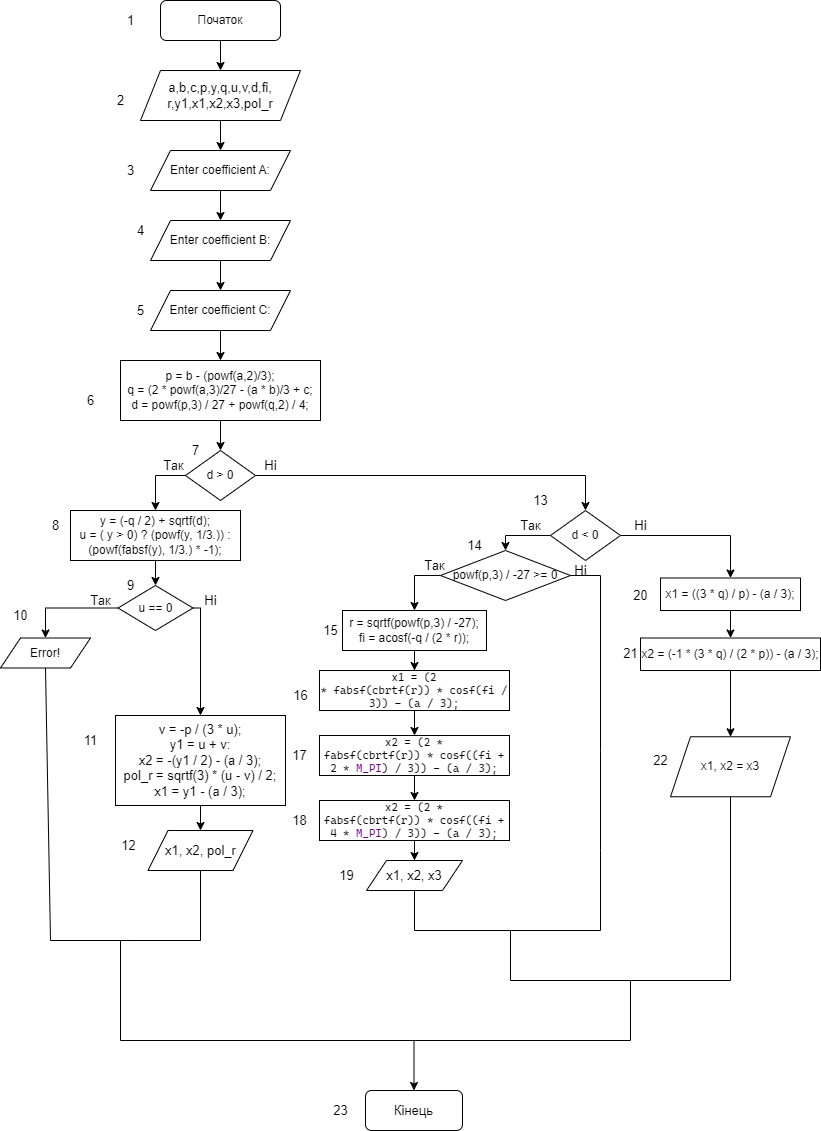
printf("\nx1 = %.4f\n", x1);

printf("x2 = x3 = %.4f", x2);

return 1;

}

}

**Блок схема до програми**

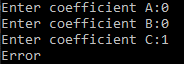
Введені та одержані результати

Зображення, що містить текст

Автоматично згенерований опис1.

Зображення, що містить текст

Автоматично згенерований опис2.

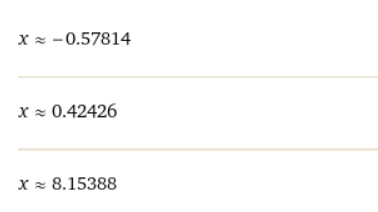
3.



Зображення, що містить текст

Автоматично згенерований опис1.

2.



3. Рівняння з заданими коефіцієнтами не є кубічним тому програма не здатна знайти корені.

Висновок: Тестові приклади для перевірки коректності розрахунків відповідають отриманим. Програма працює коректно. Програма вирішує поставлене завдання.