**Лабораторна робота №5.**

Програмування найпростіших обчислювальних алгоритмів лінійної структури

**Варіант № 19**

**Мета роботи**: навчитися складати і програмувати мовою С алгоритми з розгалуженнями, використовуючи послідовну перевірку умов, вкладену перевірку умов, а також оператор вибору.

**Завдання**: для обчислення значень заданої логічною залежністю функції

розробити **два алгоритми**:

1)**з** **послідовною** **перевіркою** **умов** (неповна форма умовного оператора); видачу повідомлень про помилки здійснювати зразу ж при їхньому виявленні, а видачу результату обчислення значення функції f(x) —

у кінці програми; 2) **з вкладеною перевіркою умов**; порівняння в умовах не повторювати; видачу повідомлень про помилки і видачу результату обчислення значення функції f(x) здійснювати в кінці програми.

Намалювати блок-схеми й написати програми обчислення значень заданої логічною залежністю функції f(x) при довільних значеннях параметрів а і b і незалежної змінної х. В алгоритмах передбачити перевірку, чи немає ділення на нуль, чи підкореневий вираз невід’ємний, чи аргумент функції логарифма набуває додатних значень тощо. Вважається, що функція визначена на заданому проміжку, а поза ним не визначена. Коментарі в програмі обов’язкові. За алгоритмами провести розрахунки не менш ніж з п’ятьма різними наборами вхідних даних, які передбачають виконання різних гілок алгоритму

Текст програми №1:

/\* Пузенко Артем Андрійович, група ТР-23, варіант №19.

Лабораторна робота № 5.1. Послідовна перевірка умов \*/

#include <stdio.h> /\*Використовуються printf(), scanf()\*/

#include <math.h> /\*Використовується system()\*/

#include <stdlib.h> /\*Використовуються sin(), sqrt(), fabs()\*/

int main(void)

{

/\* Обчислення значення функції, заданої логічною залежністю.

Використання послідовних умовних операторів \*/

double x, // значення аргумента

y, // значення функції

a, b; // параметри функції

int ind; // індикатор друку значення функції

system("chcp 1251 & cls"); // українізація повідомлень

/\* Виведення вигляду функції на екран \*/

printf(" {|ax+7x^3| при x (-inf,9)\n");

printf("f(x)={5(bx+1)^0.5 при x [9;15)\n");

printf(" {sin(x) при x = 15\n");

/\* Введення вхідних даних \*/

printf("\nВведіть значення a = ");

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

printf("Введіть значення b = ");

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

printf("Введіть значення x = ");

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

ind = 0;

/\* Обчислення значення функції \*/

/\* Перший проміжок\*/

if (x < 9)

{

y = fabs(x \* (a + 7 \* x \* x)); ind = 1;

}

/\* Другий проміжок. ОДЗ не порушено \*/

if ((x >= 9) && (x < 15) && (b \* x + 1 >= 0))

{

y = 5 \* sqrt(b \* x + 1); ind = 1;

}

/\* Другий проміжок. ОДЗ порушено \*/

if ((x >= 9) && (x < 15) && (b \* x + 1 < 0))

{

printf("Підкореневий вираз менше 0\n");

}

/\* Третій проміжок. Особливостей нема \*/

if (x == 15)

{

y = sin(x); ind = 1;

}

/\* Значення х задано поза проміжком \*/

if (x > 15)

{

printf("Функція не визначена");

}

/\* Друк результату обчислення \*/

if (ind)

printf("f(%4.2f)=%6.3f", x, y);

printf("\n\n");

/\* Затримка вікна виведення \*/

system("pause");

return 0;

}

початок

виведення вигляду функції

введення значень a, b, x

ind=0

# x < 9

ні

так

y= fabs(a\*x+7\*x\*x\*x)

ind=1

1

1

<0

ні

так

Модуль від'ємний

# 9 ≤ x < 15

ні

так

y = 5\*sqrt(b\*x+1)

ind=1

2

2

(x >= 9) && (x < 15) && (b\*x+1 < 0)

ні

так

Підкореневий вираз менше 0

# x==15

ні

так

y = sin(x)

ind=1

ind

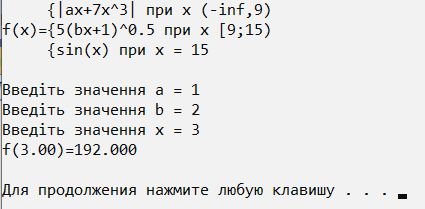
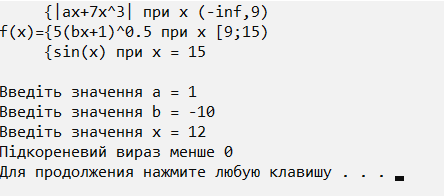
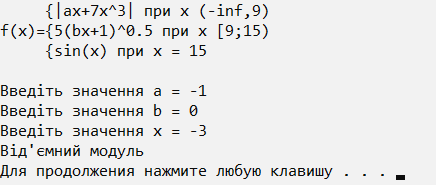
так

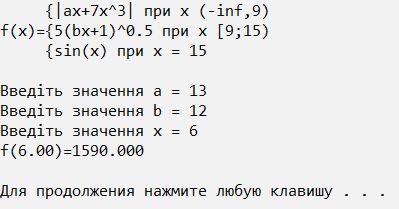
ні

виведення значення y

кінець

Приклади роботи програми №1:





Текст програми №2:

/\* Пузенко Артем Андрійович, ТР-23, варіант №19.

Лабораторна робота № 5.2. Вкладена перевірка умов \*/

#include <stdio.h> /\*Використовуються printf(), scanf()\*/

#include <stdlib.h> /\*Використовується system()\*/

#include <math.h> /\*Використовуються sin(), sqrt(), fabs()\*/

int main(void)

{

/\* Обчислення значення функції, заданої логічною залеж-

\* ністю. Використання вкладених умовних операторів. \*/

double x, // значення аргумента

y, // значення функції

a, b; // параметри функції

int nom; // номер повідомлення (0 – значення обчислено)

system("chcp 1251 & cls"); // українізація повідомлень

/\* Виведення вигляду функції на екран \*/

printf(" {|ax+7x^3| pry x (-inf,9)\n");

printf("f(x)={5(bx+1)^0.5 pry x [9;15)\n");

printf(" {sin(x) pry x=15\n");

/\* Введення вхідних даних \*/

printf("\nВведіть значення a = ");

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

printf("Введіть значення b = ");

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

printf("Введіть значення x = ");

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

nom = 0;

/\* Обчислення значення функції \*/

if (x < 9) // Перший проміжок

y = fabs(x \* (a + 7 \* x \* x)); // Особливостей нема

else

if (x < 15) // Другий проміжок

if (b \* x + 1 < 0) nom = 1; // ОДЗ порушено

else y = 5 \* sqrt(b \* x + 1); // ОДЗ не порушено

else

if (x == 15) // Третій проміжок - особливостей нема

y = sin(x);

else nom = 2; // Значення х задано поза проміжком

/\* Друк результату обчислення \*/

switch (nom) {

case 1: printf("Підкореневий вираз менше 0"); break;

case 2: printf("Функція не визначена"); break;

default: printf("f(%4.2f)=%6.3f", x, y);

}

/\* Затримка вікна виведення \*/

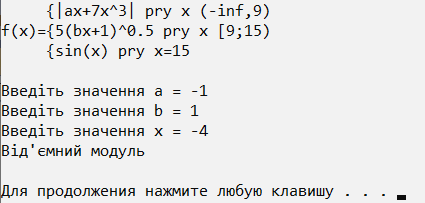
printf("\n\n");

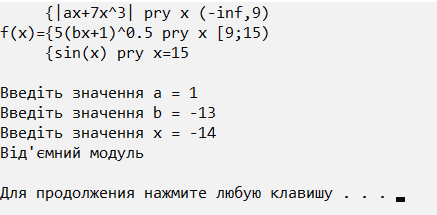
system("pause");

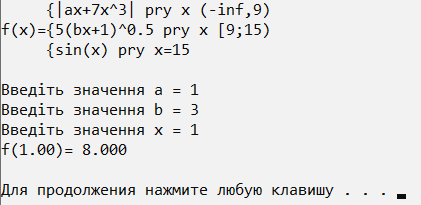
return 0;

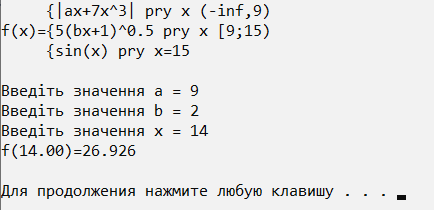
}

Приклади роботи програми №2:









Блок схема алгоритму №2:

початок

Виведення вигляду функції

Введення значень a, b, x

Nom=0

1

1

X < 9

ні

так

y = fabs(a \* x + 7 \* x \* x \* x)

(x<15)&& (b \* x + 1 >= 0)

ні

так

(x == 15)

y = 5 \* sqrt(b \* x + 1);

ні

так

nom=1

y = sin(x)

# nom

Виведення значення y

0

1

Функція не визначена

2

2

кінець

**Висновок**: результаті виконання лабораторної роботи на тему “Реалізація алгоритмів з розгалуженнями” я розробила два алгоритми й подала їх за допомогою блок-схем, а також реалізувала ці алгоритми у вигляді програм мовою С. Відповідно до одного алгоритму видача повідомлень про помилки, здійснюється зразу ж при їхньому виявленні, а видача обчисленого значення функції — у кінці програми. Відповідно до другого алгоритму вся видача результатів здійснюється в кінці програми. За програмами провела розрахунки на різних наборах даних

Розроблений алгоритм з використанням послідовних умовних операторів має простішу структуру і легший для сприйняття й реалізації, але він менш ефективний, ніж другий алгоритм з вкладеними умовами. Крім того, використання оператора вибору дало можливість відокремити блок обчислень від блоку видачі результатів.