**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ**

**ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО** »

**Інститут прикладного системного аналізу**

**Кафедра математичних методів системного аналізу**

**Звіт**

**про виконання лабораторної роботи №3**

**з дисципліни «Алгоритмізація та програмування»**

Виконав:

студент I курсу, группи КА-07

Лещинський Богдан Денисович

Прийняв:

к.т.н., доцент Безносик О. Ю.

Київ – 2020

Завдання 1

Скласти блок-схему алгоритма і програму для обчислення суми довільної низки чисел.

Аналіз умови задачі

Виходячи з умови задачі, можна сказати, що алгоритм задачі буде нелінійним, оскільки буде застосовуватися цикл для обчислення суми довільної низки чисел з кількістю ітерацій, що буде залежати від кількості чисел, що дадаються. Вхідних змінних - дві: n, х. Вихідна змінна - одна: sum. Тип данних вхідної змінної n - натуральні числа, х, sum - дійсни числа подвійної точності (double).

Блок схема алгоритму для завдання 1 представлена на рисунку 1.1:

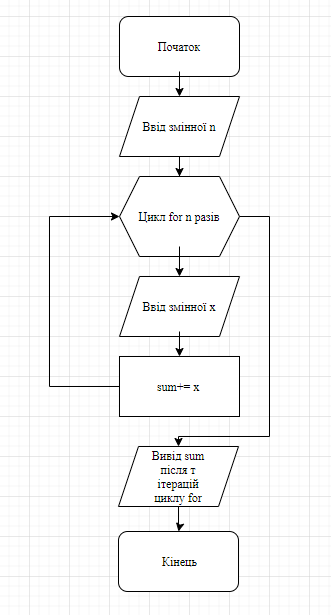


Рис. 1.1

Код программи:

#include<stdio.h>

int main(){

int n;

double x;

double sum = 0;

printf("Enter a number of operands you want to add: \n");

scanf("%d",&n);

printf("Enter operands in the entered quantity one after another: \n");

for(int i = 0; i<n ; i++){

scanf("%lf",&x);

sum+=x;

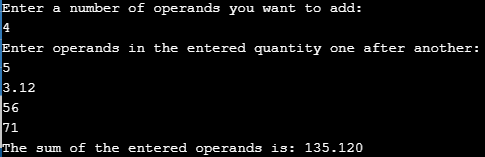
}

printf("The sum of the entered operands is: %.3lf",sum);

return 0;

}

Результат роботи програми представлентй на рисунку 1.2:



Завдання 2

Скласти блок-схему алгоритма і програму для обчислення добутку.

Аналіз умови задачі

Виходячи з умови задачі, можна сказати, що алгоритм задачі буде нелінійним, оскільки для обчислення добутку буде застосовуватися цикл з розгалудженнями всередині нього. Вхідних змінних - дві: n, х. Вихідна змінна - одна: sum. Тип данних вхідної змінної n - натуральні числа, х, sum - дійсні числа подвійної точності (double).

Формула для розрахунків



Блок схема алгоритму для завдання 2 представлена на рисунку 2.1:

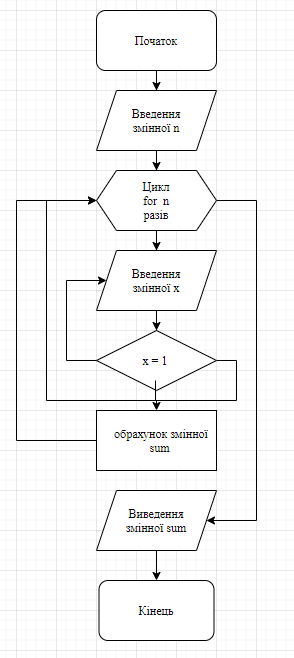


Рис. 2.1

Код программи:

#include<stdio.h>

int main(){

int n;

double x,sum;

sum = 1;

printf("Enter a number of operands you want to compute: \n");

scanf("%d",&n);

printf("Enter operands(other than 1) in the entered quantity one after another: \n");

for(int i = 1; i<=n ; i++){

scanf("%lf",&x);

while(x==1){

printf("Uncomputable for x=1, try entering x other than 1: \n");

scanf("%lf",&x);

}

sum \*= (x\*x)/(1-x);

}

printf("The output of the entered operands: %.4lf", sum);

return 0;

}

Результат роботи програми представлентй на рисунку 2.2:

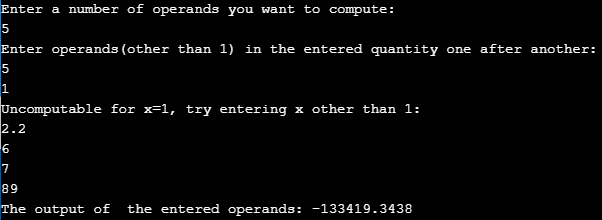


Рис. 2.2

Завдання 3

Скласти блок-схему алгоритма і програму для обчислення числа сполучень з m чисел по n.

Аналіз умови задачі

Виходячи з умови задачі, можна сказати, що алгоритм задачі буде нелінійним, оскільки для обчислення числа сполучень з m чисел по n потрібно буде ввечти перевірку чи n<m, отже доведеться вобити розгалуження. Вихідних змінних - дві: n, m. Вихідна змінна - одна: С. Тип данних вхідних змінних - натуральні числа. Тип данних вихідної змінної - натуральні числа.

Формула для розрахунків



Блок схема алгоритму для завдання 3 представлена на рисунку 3.1:

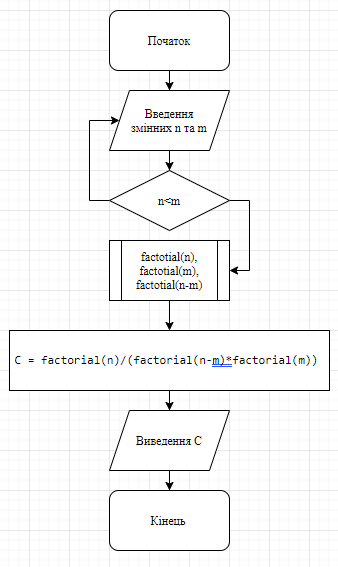


Рис. 3.1

Код программи:

#include<stdio.h>

int factorial(int a){

int factA;

if(a==1){

factA = a;

return factA;

}

else{

factA = a;

return factA\*factorial(factA-1);

}

}

int main(){

int n,m,C;

printf("Enter natural values of n and m respectively to compute C(n,m): \n");

scanf("%d %d",&n,&m);

while(n<m){

printf("Uncomputable, try entering values of n and m so that n>m: \n");

scanf("%d %d",&n,&m);

}

C = factorial(n)/(factorial(n-m)\*factorial(m));

printf("C(n,m) for entered n and m is: %d",C);

return 0;

}

Результат роботи програми представлентй на рисунку 3.2:

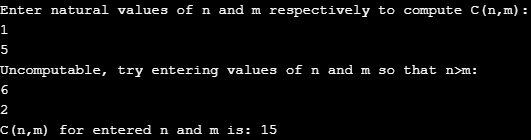


Рис. 3.2

Завдання 4

Скласти програму пошуку мінімального елемента з низки довільних чисел N, що вводяться з клавіатури.

Аналіз умови задачі

Виходячи з умови задачі, можна сказати, що алгоритм задачі буде нелінійним, оскільки буде застосовуватися цикл для пошуку мінімального елемента з низки довільних чисел N. Вихідних змінних - три: minimal, x, n. Вихідна змінна - одна : minimal. Тип динних вхідних змінних x, minimal - дійсні числа подвійної точності (double), n - натуральні числа. Тип данних вихідної змінної - натуральні числа.

Блок схема алгоритму для завдання 4 представлена на рисунку 4.1:

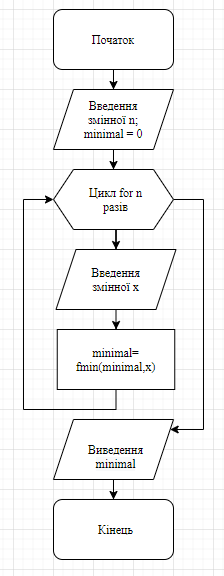


Рис. 4.1

Код программи:

#include<stdio.h>

#include<math.h>

int main(){

int n;

double minimal,x;

minimal = 0;

printf("Enter from how many numbers a minimal element must be found: \n");

scanf("%d",&n);

printf("Enter numbers in entered quantity one after another: \n");

for(int i = 1; i<=n; i++){

scanf("%lf",&x);

minimal = fmin(minimal,x);

}

printf("The minimal element among the entered number is: %.2lf",minimal);

return 0;

}

Результат роботи програми представлентй на рисунку 4.2:

Завдання 5

Скласти схему алгоритму і програму обчислень суми ряду з точністю eps1 = 0.01 і eps2 = 0.0001.Вивести в двох випадках на друк значення s, n (число ітерацій), еps, а також лістинг програми. Зробити висновки.

Аналіз умови задачі