МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпропетровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Технічна Кібернетика»

**Лабораторна робота №3**

**з дисципліни «Основи програмування»**

**на тему: «**Розробка програм з циклічною обробкою даних.**»**

Виконав:

студент гр.ПЗ1911

Сафонов Д.Є.

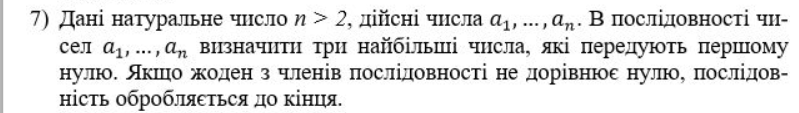
Прийняла:

Нежуміра О.И.

Дніпро, 2019

**Тема.** Розробка програм з циклічною обробкою даних.

**Мета.** Вивчити типи операторів циклу. Навчитися використовувати різні типи циклів для розв’язання задач.



Розробити три програми відповідно до обраного варіанту завдання на основі трьох видів циклів.

## **Вимоги до програми:**

- вхідні дані вводяться з клавіатури;

- передбачити перевірку вхідних даних на відповідність діапазону значень і некоректні символи;

- результати роботи програми виводяться на екран;

- *забороняється використовувати*[*масиви*](http://lider.diit.edu.ua/mod/lesson/view.php?id=40269)*(!!!).*

## **Вимоги до тексту програми:**

- коментарі щодо призначення програми, її вхідних і вихідних даних;

- коментарі щодо призначення кожного блоку програми, дій окремих операторів для пояснення алгоритму;

- самодокументованість коду: всі ідентифікатори повинні мати назви, що відповідають суті змінних.

## **1) зовнішні специфікації програми;**

*1.1)вхідні дані*

|  |  |  |  |
| --- | --- | --- | --- |
| Найменування даних | Умовне позначення | Вимоги до данних | Приклад |
| Запланована кількість чисел у послідовності | n | натуральне число, більше або дорівнює трьом | 4 |
| Перше число послідовності, після сортировки найбільше | L | натуральне число або нуль | 0 |
| Друге число послідовності, після сортировки середне | M | натуральне число або нуль | 1 |
| Трете число послідовності, після сортировки найменьше | S | натуральне число або нуль | 2 |
| Усі наступні числа, кандидати на S, M, L | a | натуральне число або нуль | 3 |

*1.2)вихідні дані*

|  |  |  |  |
| --- | --- | --- | --- |
| Найбільше число | L | натуральне число або нуль | 2 |
| Середне число | M | натуральне число або нуль | 1 |
| Найменьше число | S | натуральне число або нуль | 0 |

## **2) метод рішення завдання;**

1. Программа отримує значення n, L, M, S.

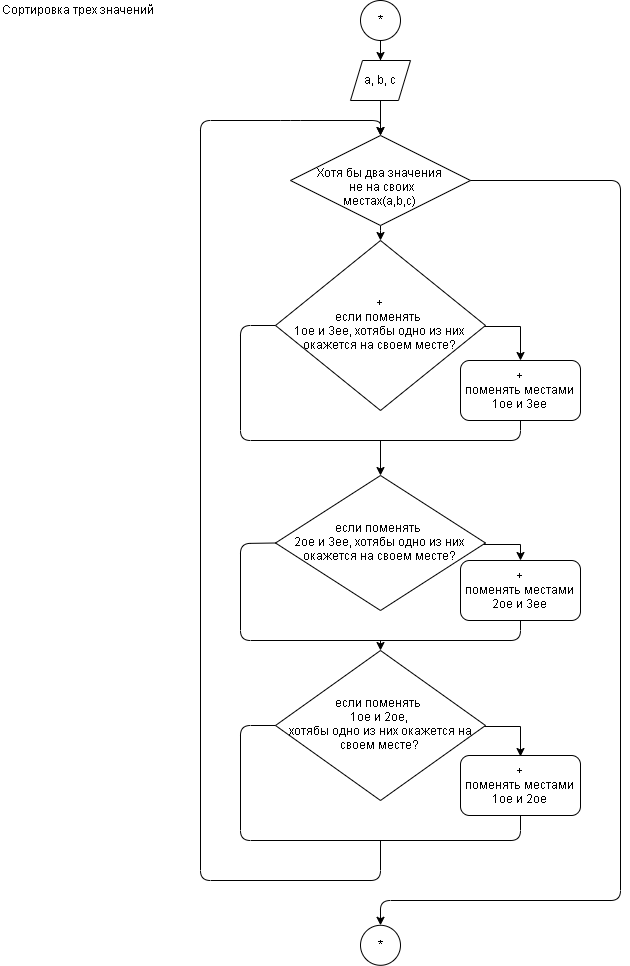
2. Сортує L, M, S, так щоб виконувалась нерівність .

3. Цикл, який отримує значення чисел , якщо якесь з чисел дорівнює нулю, то після нього введеня даних закінчуєтся. Якщо введене число більше за якесь з L, M, N, введене число та найбільше з L, M, N, але меньше за введене міняються значеннями, потім знову порівняння, доки усі три числа L, M, N більші за «введене».

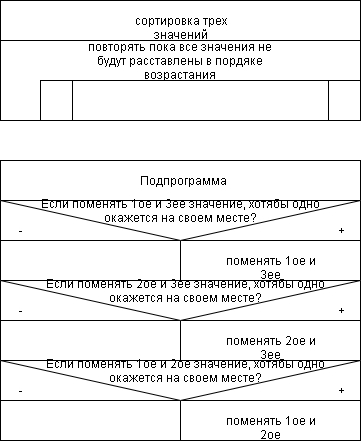
4. Виведення L, M, S.

Для цієї программи я розробив алгоритм який сортує значення трьох змінних.

Блок-схема:

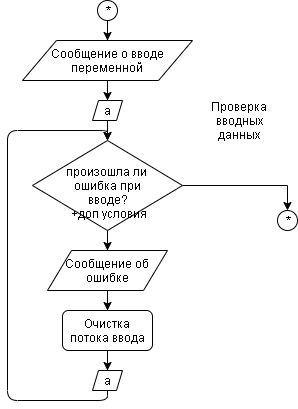


N-S

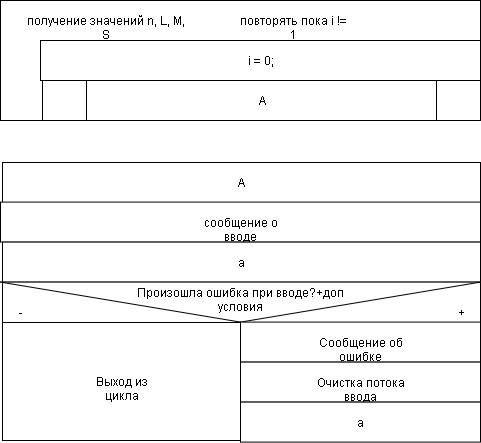


Також я зробив свій хєдер з чотирма функціями, які отримують вхідні дані та перевіряють чи належать ці дані до потрібного діапазону.

Блок-схема



N-S



## **Inp\_val.h**

//cinat - input natural number only (N)

//cinat0 - (N+0)

//cint - integers only (Z)

//cinum - numerical only (R)

#include <iostream>

#include <windows.h>

#include <cmath>

using namespace std;

int cinat(string a, string b)

{

double aaa;

cout << a << endl;

cin >> aaa;

for (bool i = 0; i != 1;)

{

if (cin.fail() || (floor(aaa) != ceil(aaa)) || (aaa < 1)) {//floor biggest int x thats smaller ot equal than a; ceil smallest int x thats bigger or equal than a; if floor == ceil, than a - int

cout << b << endl;

cin.clear();

cin.ignore();

cin >> aaa;

}

else {

i = 1;

}

}

return (int)aaa;

}

int cinat0(string a, string b)

{

double aaa;

cout << a << endl;

cin >> aaa;

for (bool i = 0; i != 1;)

{

if (cin.fail() || (floor(aaa) != ceil(aaa)) || (aaa < 0)) {//floor biggest int x thats smaller ot equal than a; ceil smallest int x thats bigger or equal than a; if floor == ceil, than a - int

cout << b << endl;

cin.clear();

cin.ignore();

cin >> aaa;

}

else {

i = 1;

}

}

return (int)aaa;

}

int cint(string a, string b)

{

double aaa;

cout << a << endl;

cin >> aaa;

for (bool i = 0; i != 1;)

{

if (cin.fail() || (floor(aaa) != ceil(aaa))) {//floor biggest int x thats smaller ot equal than a; ceil smallest int x thats bigger or equal than a; if floor == ceil, than a - int

cout << b << endl;

cin.clear();

cin.ignore();

cin >> aaa;

}

else {

i = 1;

}

}

return (int)aaa;

}

double cinum(string a, string b)

{

double aaa;

cout << a << endl;

cin >> aaa;

for (bool i = 0; i != 1;)

{

if (cin.fail()) {

cout << b << endl;

cin.clear();

cin.ignore();

cin >> aaa;

}

else {

i = 1;

}

}

return aaa;

}

## **3) набір тестів для перевірки правильності виконання завдання;**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Назва | Вхідні дані | | | | Очікувані результати | | | Додатки |
|  | n | L | M | S | L | M | S |  |
| n<3 | -2 |  |  |  | 3 | 2 | 1 | Программа попрохає ввести число н знову |
| N не ціле | 0.8057857 |  |  |  | 3 | 2 | 1 | Программа попрохає ввести число н знову |
| Порядок1 | 3 | 1 | 2 | 3 | 3 | 2 | 1 |  |
| Порядок2 | 3 | 3 | 1 | 2 | 3 | 2 | 1 |  |
| Порядок3 | 3 | 2 | 3 | 1 | 3 | 2 | 1 |  |
| Порядок4 | 3 | 3 | 2 | 1 | 3 | 2 | 1 |  |
| Порядок5 | 3 | 2 | 1 | 3 | 3 | 2 | 1 |  |
| Порядок6 | 3 | 1 | 3 | 2 | 3 | 2 | 1 |  |

## **4.1) для варіанту реалізації завдання на основі циклу з лічильником(for).**

    a) алгоритм розв’язання задачі

a1) блок-схема

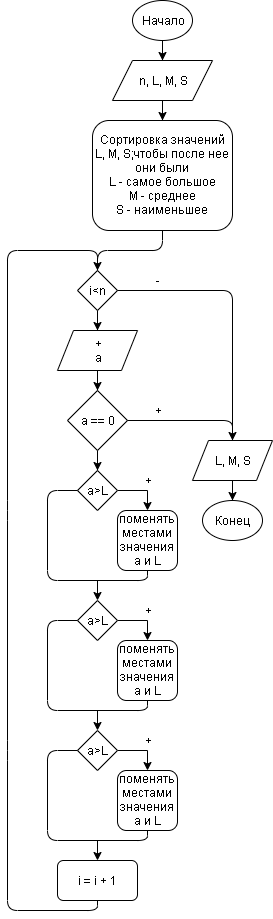


Рисунок 1

a2) діаграма Н-Ш

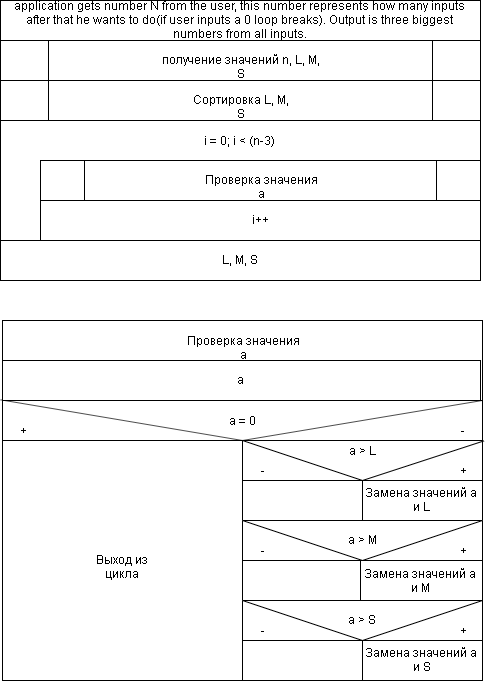


Рисунок 2

    b) текст програми;

//application gets number N from the user, this number represents how many inputs after that he wants to do(if user inputs a 0 loop breaks).

//Output is three biggest numbers from all inputs.

#include <iostream>

#include <algorithm>

#include <cmath>

#include <cstdlib>

#include "inp\_val.h"

using namespace std;

int main()

{

cout << "The application takes at least 3 natural number inputs, at most n(user specified)," << endl;

cout << "first three can be equal to 0, but if any input after 3rd will be equal to 0, input stops" << endl;

cout << "Output will be three biggest numbers from the input" << endl;

cout << endl;

#pragma region input S,M,L,n

#pragma region input n

double n;

cout << "input natural number n" << endl;

cin >> n;

for (bool i = 0; i != 1;)

{

if (cin.fail() || (floor(n) != ceil(n)) || (n < 3)) {//floor biggest int x thats smaller ot equal than a; ceil smallest int x thats bigger or equal than a; if floor == ceil, than a - int

cout << "Wrong input, try again" << endl;

cin.clear();

cin.ignore();

cin >> n;

}

else {

i = 1;

}

}

int L;

string dialogL = "input natural number a1";

string error = "Wrong input, try again";

L = cinat0(dialogL, error);

int M;

string dialogM = "input natural number a2";

M = cinat0(dialogM, error);

int S;

string dialogS = "input natural number a3";

S = cinat0(dialogS, error);

#pragma endregion

int ax;

#pragma region sort

while ((L < S) || (L < M) || (M < S)) {

if (L < S) {

ax = L;

L = S;

S = ax;//xchg L, S

}

else if (L < M) {

ax = M;

M = L;

L = ax;//xchg L, M

}

else if (M < S) {

ax = M;

M = S;

S = ax;//xchg M, S

}

}

#pragma endregion sorting a1, a2, a3 and putting them to S, M, L; so that L>M>S

for (int i = 0; i < (n - 3); i++) {

#pragma region input a

double a;

cout << "input natural number a" << i+4 << endl;

cin >> a;

for (bool i = 0; i != 1;)

{

if (cin.fail() || (floor(a) != ceil(a)) || (a < 0)) {//floor biggest int x thats smaller ot equal than a; ceil smallest int x thats bigger or equal than a; if floor == ceil, than a - int

cout << "Wrong input, try again" << endl;

cin.clear();

cin.ignore();

cin >> a;

}

else {

i = 1;

}

}

#pragma endregion

#pragma region swaps

if (a == 0) {

break;

}

if (a > L) {

ax = L;

L = a;

a = ax;

}

if (a > M) {

ax = M;

M = a;

a = ax;

}

if (a > S) {

S = a;

}

#pragma endregion

}

std::cout << L << ">" << M << ">" << S << endl;

std::system("pause");

return 0;

}

    c) результати тестування програми та їх аналіз;

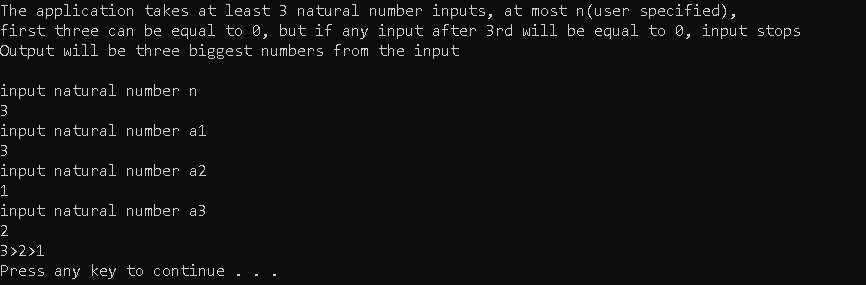


Рисунок 3

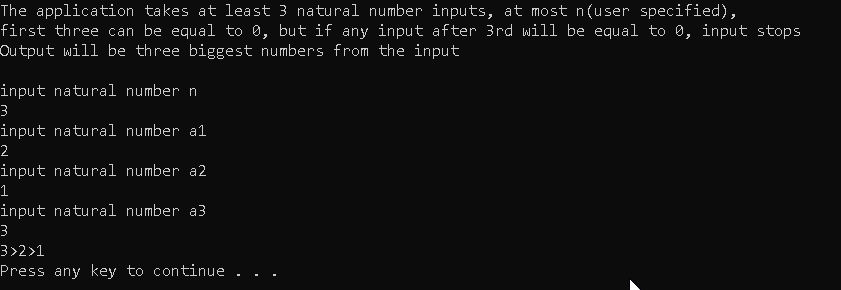


Рисунок 4

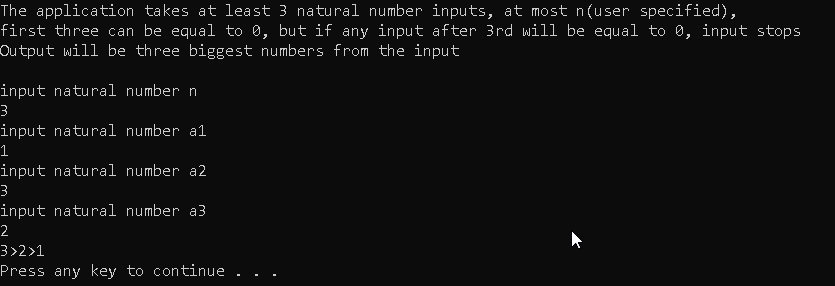


Рисунок 5

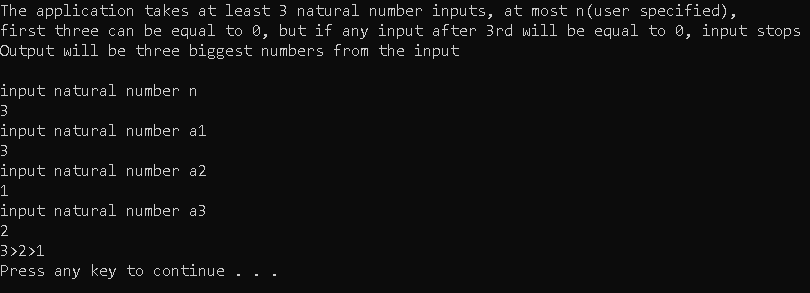


Рисунок 6

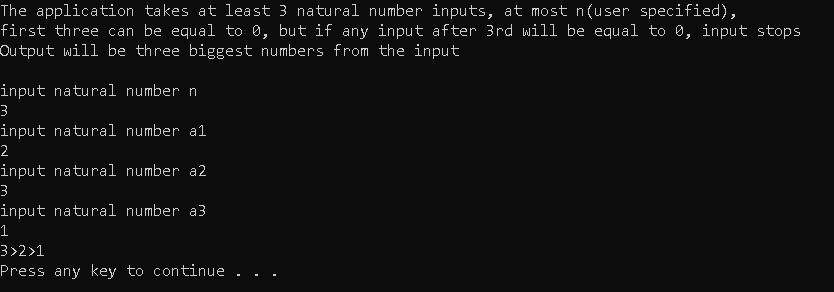


Рисунок 7

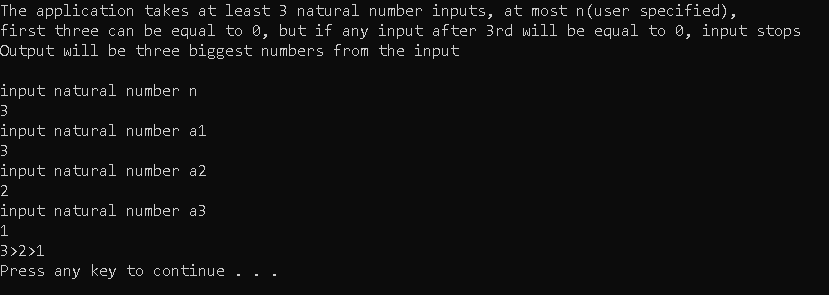


Рисунок 8

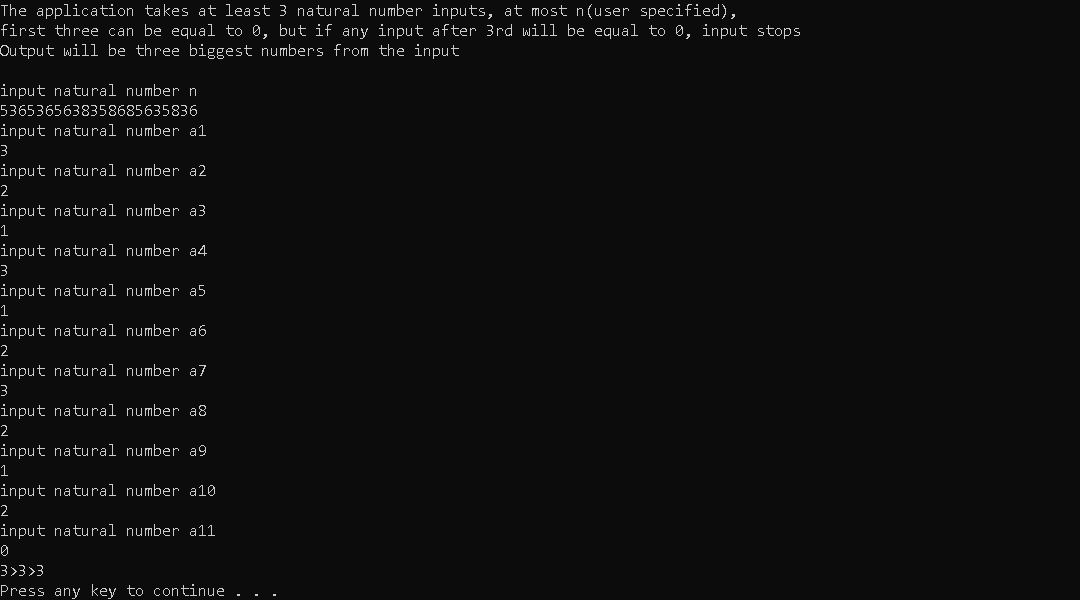


Рисунок 9

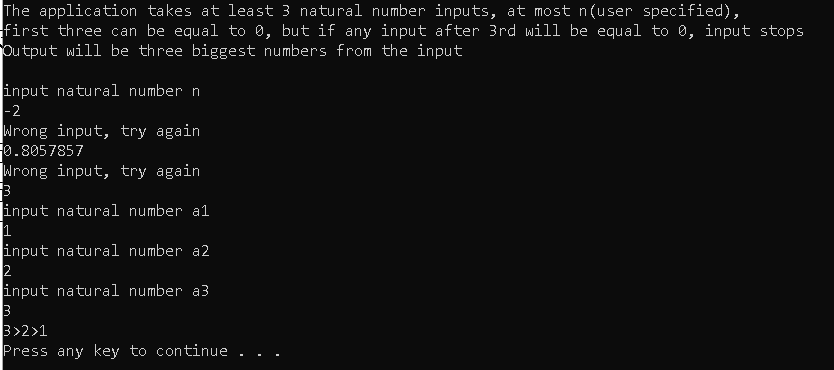


Рисунок 10

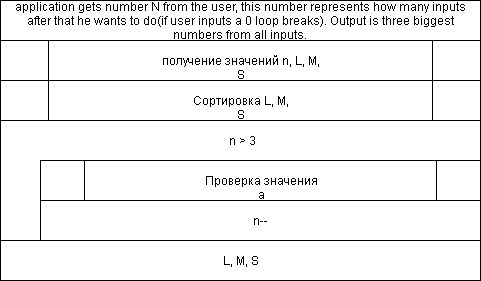
Программа працює корректно, усі спеціфікації збігаються.

## **4.2) для варіанту реалізації завдання на основі циклу з передумовою(while).**

        a) алгоритм розв’язання задачі

a1) блок-схема

a2) та діаграма Н-Шs



    b) текст програми;

//application gets number N from the user, this number represents how many inputs after that he wants to do(if user inputs a 0 loop breaks).

//Output is three biggest numbers from all inputs.

#include <iostream>

#include <algorithm>

#include <cmath>

#include <cstdlib>

#include "inp\_val.h"

using namespace std;

int main()

{

cout << "The application takes at least 3 natural number inputs, at most n(user specified)," << endl;

cout << "first three can be equal to 0, but if any input after 3rd will be equal to 0, input stops" << endl;

cout << "Output will be three biggest numbers from the input" << endl;

cout << endl;

#pragma region input S,M,L,n

int n;

string dialogN = "input natural number n";

string error = "Wrong input, try again";

n = cinat(dialogN, error);

int L;

string dialogL = "input natural number a1";

L = cinat0(dialogL, error);

int M;

string dialogM = "input natural number a2";

M = cinat0(dialogM, error);

int S;

string dialogS = "input natural number a3";

S = cinat0(dialogS, error);

#pragma endregion

int ax;

#pragma region sort

while ((L < S) || (L < M) || (M < S)) {

if (L < S) {

ax = L;

L = S;

S = ax;//xchg L, S

}

else if (L < M) {

ax = M;

M = L;

L = ax;//xchg L, M

}

else if (M < S) {

ax = M;

M = S;

S = ax;//xchg M, S

}

}

#pragma endregion sorting a1, a2, a3 and putting them to S, M, L; so that L>M>S

int i = 0;

while (n > 3) {

#pragma region input a

double a;

cout << "input natural number a" << i+4 << endl;

cin >> a;

for (bool i = 0; i != 1;)

{

if (cin.fail() || (floor(a) != ceil(a)) || (a < 0)) {//floor biggest int x thats smaller ot equal than a; ceil smallest int x thats bigger or equal than a; if floor == ceil, than a - int

cout << "Wrong input, try again" << endl;

cin.clear();

cin.ignore();

cin >> a;

}

else {

i = 1;

}

}

#pragma endregion

#pragma region swaps

if (a == 0) {

break;

}

if (a > L) {

ax = L;

L = a;

a = ax;

}

if (a > M) {

ax = M;

M = a;

a = ax;

}

if (a > S) {

S = a;

}

#pragma endregion

n -= 1;

i += 1;

}

std::cout << L << ">" << M << ">" << S << endl;

std::system("pause");

return 0;

}

    c) результати тестування програми та їх аналіз;

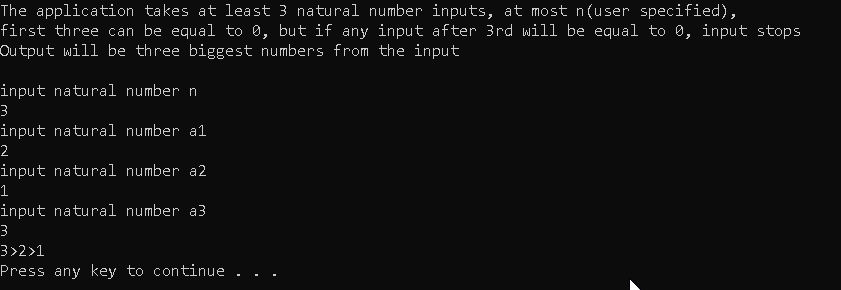


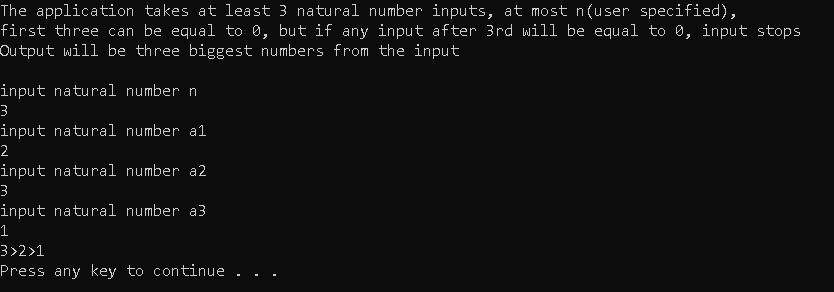
Рисунок 11

Рисунок 12

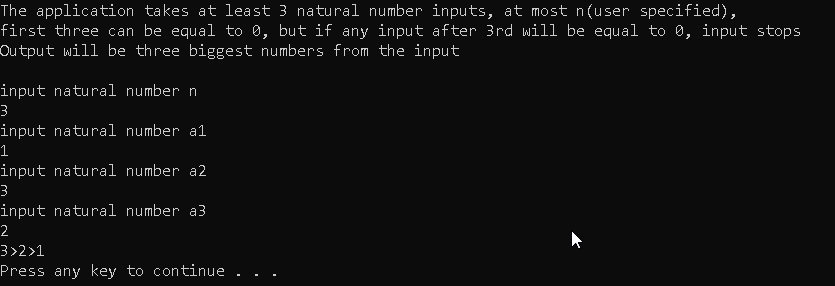


Рисунок 13

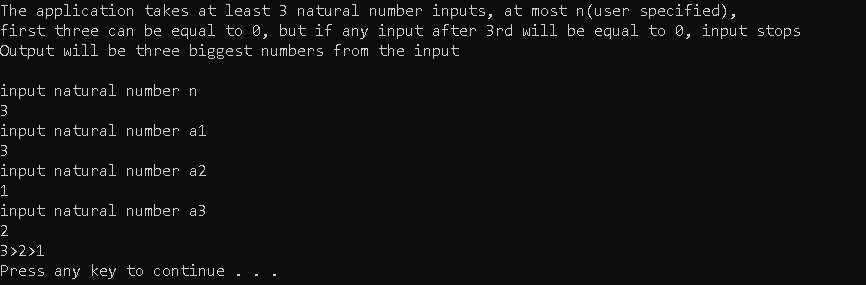


Рисунок 14

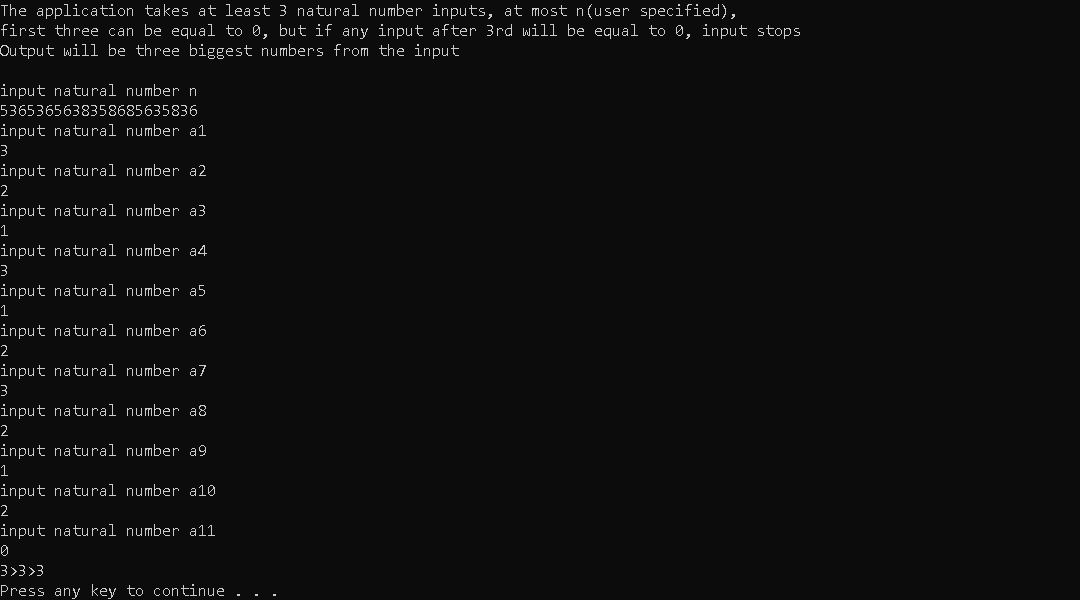


Рисунок 15

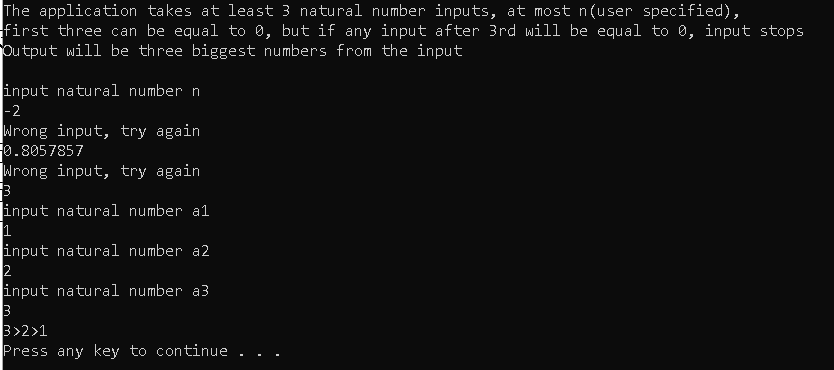


Рисунок 16

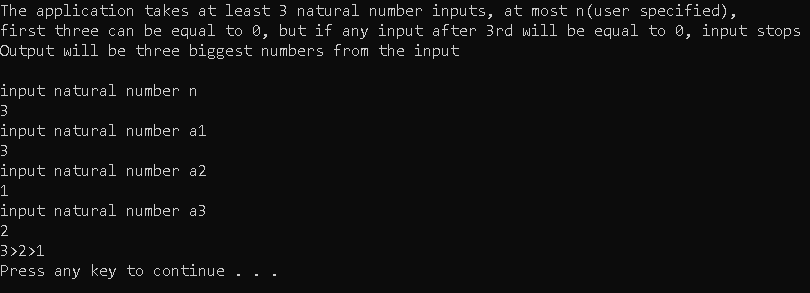


Рисунок 17

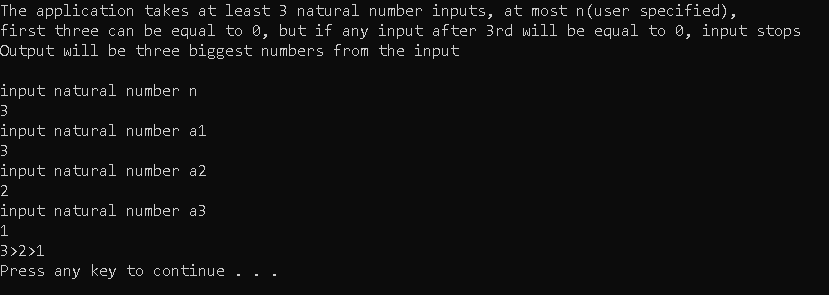


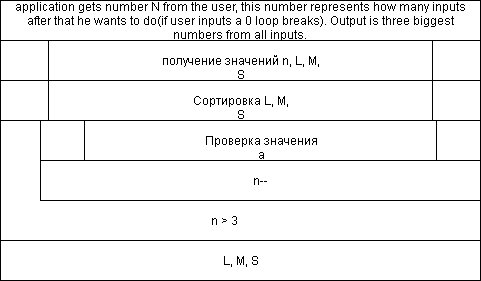
Рисунок 18

## **4.3) для варіанту реалізації завдання на основі циклу з постумовою(do while).**

    a) алгоритм розв’язання задачі

a1) блок-схема

a2) та діаграма Н-Ш



b) текст програми;

//application gets number N from the user, this number represents how many inputs after that he wants to do(if user inputs a 0 loop breaks).

//Output is three biggest numbers from all inputs.

#include <iostream>

#include <algorithm>

#include <cmath>

#include <cstdlib>

#include "inp\_val.h"

using namespace std;

int main()

{

cout << "The application takes at least 3 natural number inputs, at most n(user specified)," << endl;

cout << "first three can be equal to 0, but if any input after 3rd will be equal to 0, input stops" << endl;

cout << "Output will be three biggest numbers from the input" << endl;

cout << endl;

#pragma region input S,M,L,n

int n;

string dialogN = "input natural number n";

string error = "Wrong input, try again";

n = cinat(dialogN, error);

int L;

string dialogL = "input natural number a1";

L = cinat0(dialogL, error);

int M;

string dialogM = "input natural number a2";

M = cinat0(dialogM, error);

int S;

string dialogS = "input natural number a3";

S = cinat0(dialogS, error);

#pragma endregion

int ax;

#pragma region sort

while ((L < S) || (L < M) || (M < S)) {

if (L < S) {

ax = L;

L = S;

S = ax;//xchg L, S

}

else if (L < M) {

ax = M;

M = L;

L = ax;//xchg L, M

}

else if (M < S) {

ax = M;

M = S;

S = ax;//xchg M, S

}

}

#pragma endregion sorting a1, a2, a3 and putting them to S, M, L; so that L>M>S

int i = 0;

do {

#pragma region input a

double a;

cout << "input natural number a" << i + 4 << endl;

cin >> a;

for (bool i = 0; i != 1;)

{

if (cin.fail() || (floor(a) != ceil(a)) || (a < 0)) {//floor biggest int x thats smaller ot equal than a; ceil smallest int x thats bigger or equal than a; if floor == ceil, than a - int

cout << "Wrong input, try again" << endl;

cin.clear();

cin.ignore();

cin >> a;

}

else {

i = 1;

}

}

#pragma endregion

#pragma region swaps

if (a == 0) {

break;

}

if (a > L) {

ax = L;

L = a;

a = ax;

}

if (a > M) {

ax = M;

M = a;

a = ax;

}

if (a > S) {

S = a;

}

#pragma endregion

n -= 1;

i += 1;

} while (n > 3);

std::cout << L << ">" << M << ">" << S << endl;

std::system("pause");

return 0;

}

    c) результати тестування програми та їх аналіз;

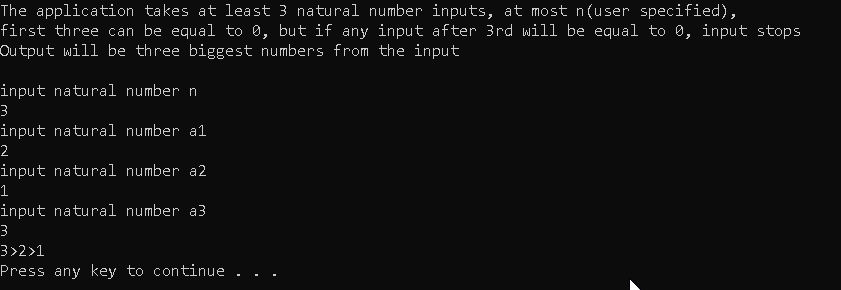


Рисунок 19

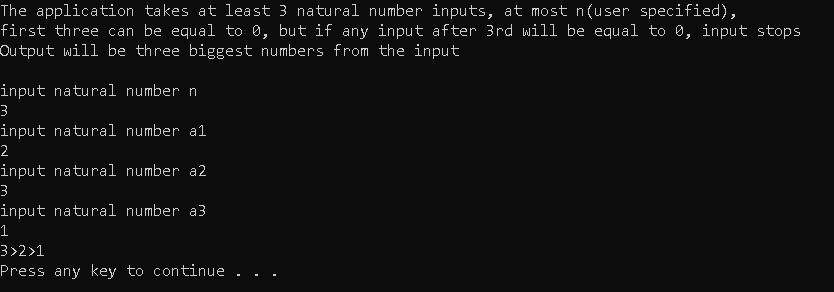


Рисунок 20

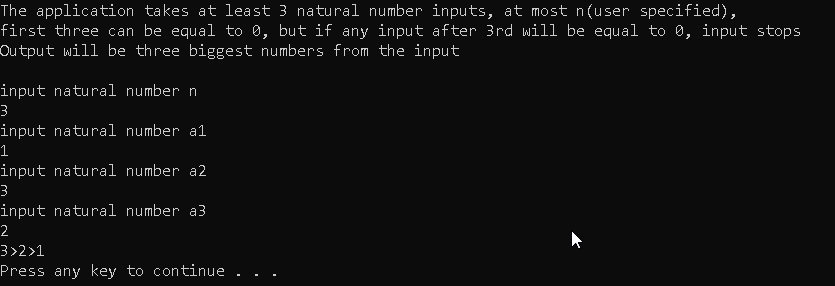


Рисунок 21

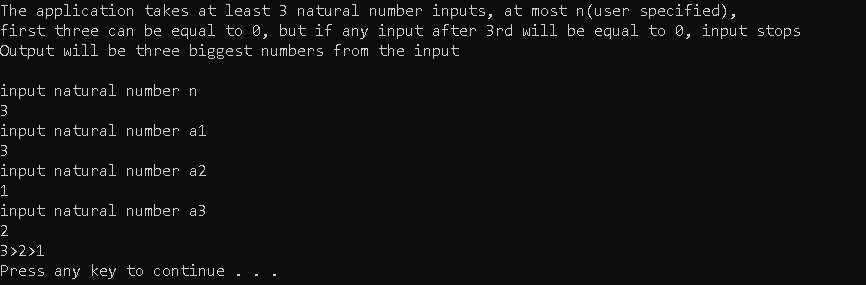


Рисунок 22

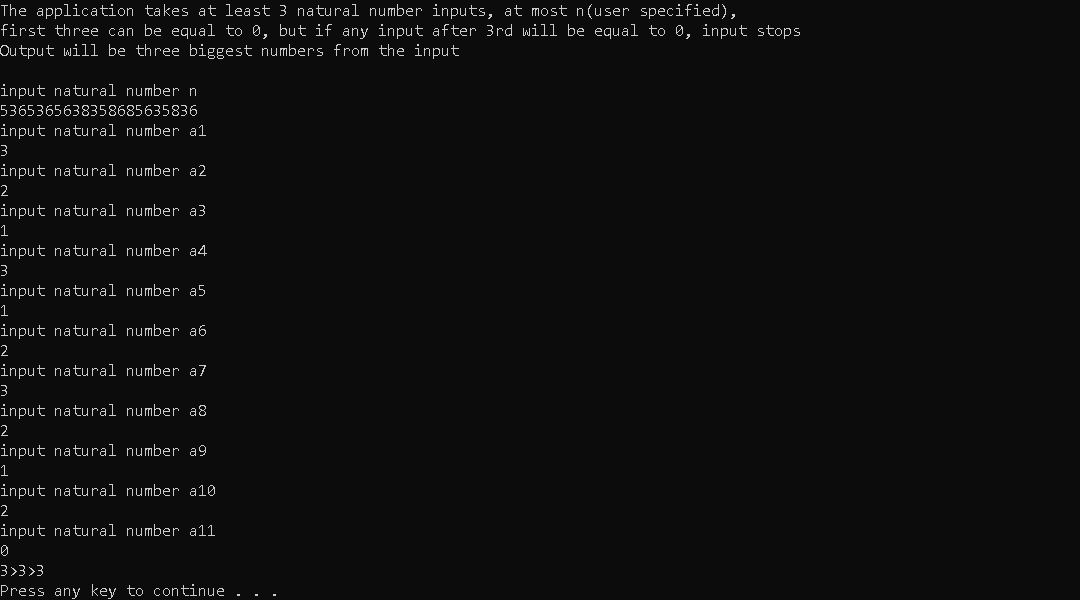


Рисунок 23

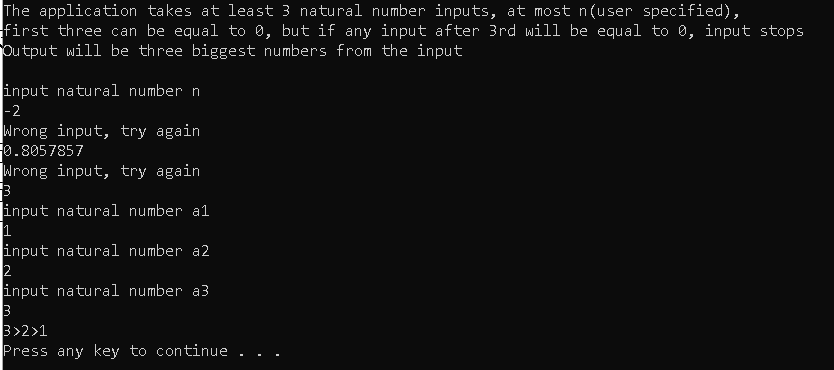


Рисунок 24

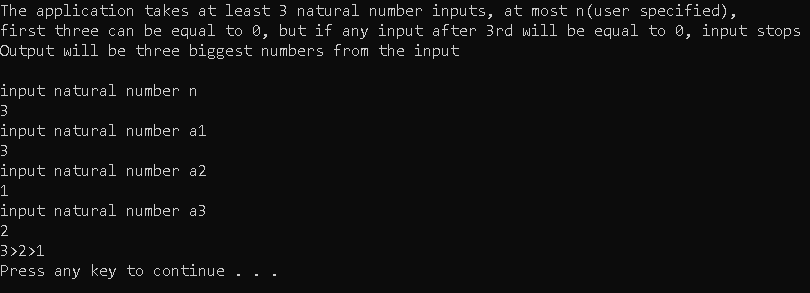


Рисунок 25

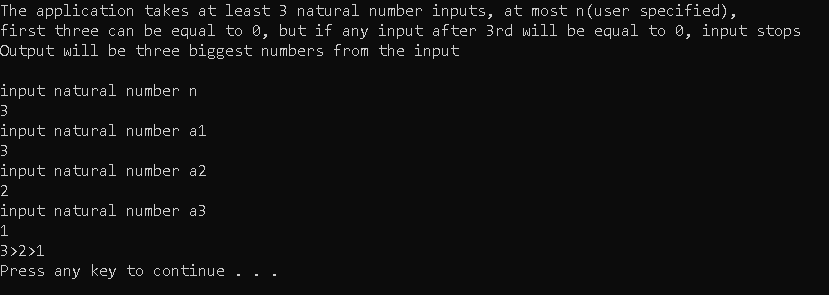


Рисунок 26

## **5) висновки щодо використання різних типів циклів в реалізації індивідуального завдання.**

For – цикл з лічильником використовується, коли потрібно півторити цикл н разів.

While – цикл з передумовою, спочатку перевірка, потім тіло циклу; використовується, коли треба получити щось конкретне на виході з циклу.

Do while – цикл з післяумовою, спочатку тіло, потім перевірка; використовується, коли треба, щоб цикл виконався хочаб один раз.

Цикли while та for можна взаємозамінювати.