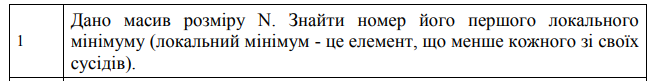
**ЛАБОРАТОРНА РОБОТА № 12**

**Дослідження роботи одновимірних масивів**

**Мета:** набуття навичок роботи з одновимірними масивами

**Хід роботи:**

**Завдання 1.** Написати та налагодити програму:



Лістинг программи:

#include <iostream>

#include <math.h>

#include <windows.h>

#include <time.h>

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(0));

int arrN[1000], count, i, a, b, indexLOCALMIN, variable, countLOCALMIN = 0;

printf("Завдання 1:\nДано масив розміру N. Знайти номер його першого локального мінімуму(локальний мінімум - це елемент, що менше кожного зі своїх сусідів).\nРозв'язання:\n");

printf("Введіть цілий розмір масиву N:"); scanf\_s("%d", &count);

printf("Введіть початкове значення масиву a (ціле):"); scanf\_s("%d", &a);

printf("Введіть кінцеве значення масиву b (ціле):"); scanf\_s("%d", &b);

if (a == b) {

printf("Помилка. Значення а не > b.");

}

else {

for (i = 0; i < count; i++) {

arrN[i] = a + rand() % (b - a + 1);

printf("%d ", arrN[i]);

}

for (i = 0; i < count; i++) {

if (arrN[i] < arrN[i - 1] && arrN[i] < arrN[i + 1]) {

variable = arrN[i];

indexLOCALMIN = i;

countLOCALMIN++;

printf("\nЛокальний мінімум %d = %d\n", countLOCALMIN, variable);

printf("Індекс локального мінімуму = %d", indexLOCALMIN);

}

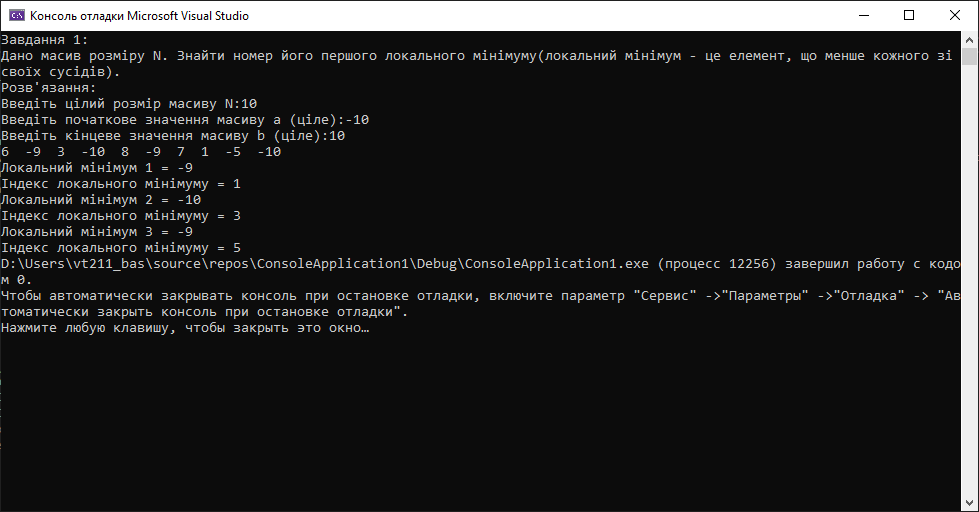
}

}

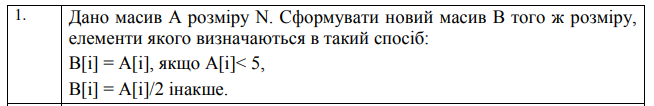
return(0);

}

Результат програми:



**Завдання 2.** Побудувати блок-схему, написати та налагодити програму. Написати тест до програми:



Лістинг программи:

#include <iostream>

#include <math.h>

#include <windows.h>

#include <time.h>

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(0));

int arrAN[1000], arrB[1000], count, i, a, b;

printf("Завдання 2:\nПобудувати блок-схему, написати та налагодити програму. Написати тест до програми.\nРозв'язання:\n");

printf("Введіть цілий розмір масиву A:"); scanf\_s("%d", &count);

printf("Введіть початкове значення масиву a (ціле):"); scanf\_s("%d", &a);

printf("Введіть кінцеве значення масиву b (ціле):"); scanf\_s("%d", &b);

if (a == b) {

printf("Помилка. Значення а не > b.");

}

else {

printf("Масив A:\n");

// Побудова масиву А

for (i = 0; i < count; i++) {

arrAN[i] = a + rand() % (b - a + 1);

printf("%d ", arrAN[i]);

}

// Підстановка елеменів у масив и елементів масиву А, які зодовільняють умові.

for (i = 0; i < count; i++) {

if (arrAN[i] < 5) {

arrB[i] = arrAN[i];

}

else {

arrB[i] = arrAN[i] / 2;

}

}

// Побудова масиву B

printf("\nНовий масив B:\n");

for (i = 0; i < count; i++) {

printf("%d ", arrB[i]);

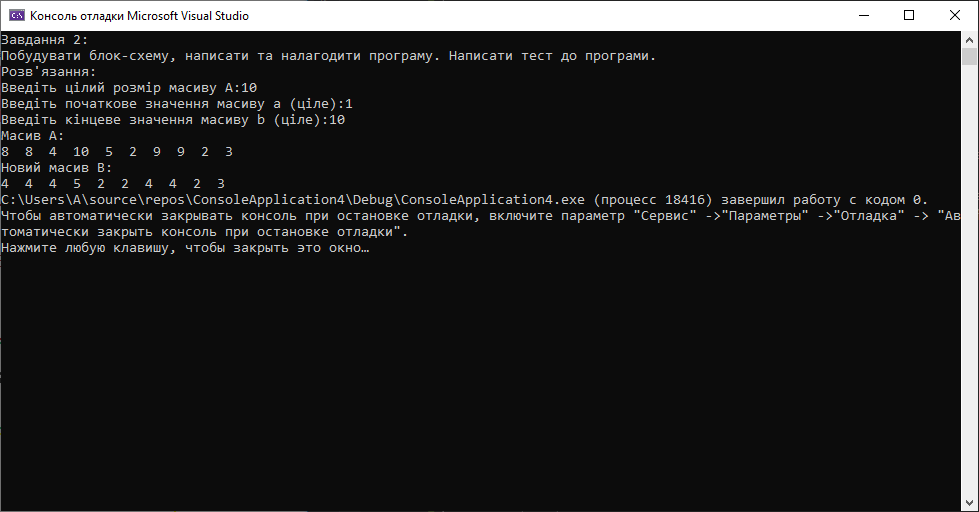
}

}

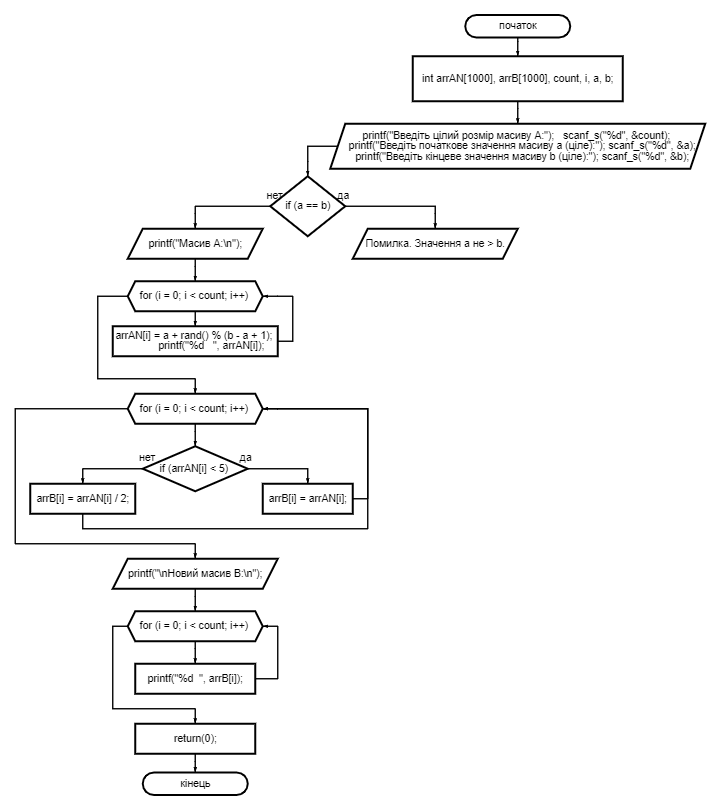
return(0);

}

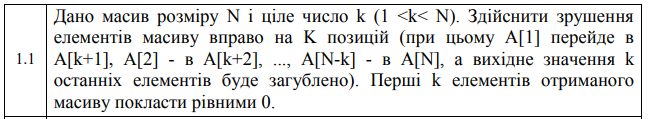
Результат програми:



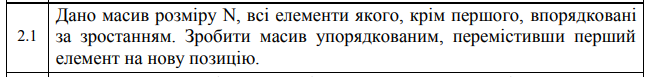
Блок схема:



**Завдання 3.** Написати програму до кожного підрозділу

1.1:

2.1:



3.1:



Лістинг програми:

1.1:

#include <iostream>

#include <math.h>

#include <windows.h>

#include <time.h>

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(0));

int arrN[1000], k, count, i, a, b, last, j;

printf("Завдання 3(1.1):\nДано масив розміру N і ціле число k (1 <k< N).\nЗдійснити елементів масиву вправо на K позицій\n(при цьому A[1] перейде в A[k + 1], A[2] - в A[k + 2], ..., A[N - k] - в A[N],\n а вихідне значення k останніх елементів буде загублено).\nПерші k елементів отриманого масиву покласти рівними 0.\n");

printf("Введіть цілий розмір масиву N:"); scanf\_s("%d", &count);

printf("Введіть початкове значення масиву a (ціле):"); scanf\_s("%d", &a);

printf("Введіть кінцеве значення масиву b (ціле):"); scanf\_s("%d", &b);

printf("Умова: (1 <k< N)\nВведіть ціле число k, яке задає зрушення позицій:"); scanf\_s("%d", &k);

if (k<1 && k<count) {

printf("Помилка. Значення k не задовільняє умові: (1 <k< N)");

}

if (a == b) {

printf("Помилка. Значення а не > b.");

}

else {

printf("Масив N(до зрушення):\n");

// Побудова масиву N

for (i = 0; i < count; i++) {

arrN[i] = a + rand() % (b - a + 1);

printf(" %d ", arrN[i]);

}

// Здійснюємо зрушення елементів на k позицій

for (int j = 1; j <= k; j++)

for (int i = count - 1; i >= 0; i--) {

if (i == count - 1) {

last = 0;

}

arrN[i] = arrN[i - 1];

if (i == 0) {

arrN[i] = 0;

}

}

// Побудова масиву зі зрушенням

printf("\nНовий масив зі зрушенням k:\n");

for (i = 0; i < count; i++) {

printf(" %d ", arrN[i]);

}

}

return(0);

}

2.1:

#include <iostream>

#include <math.h>

#include <windows.h>

#include <time.h>

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(0));

int arrN[1000], count, i, a, b, flag, temp;

printf("Завдання 3(2.1):\nДано масив розміру N, всі елементи якого, крім першого, впорядковані за зростанням.\nЗробити масив упорядкованим, перемістивши перший елемент на нову позицію.\n");

printf("Введіть цілий розмір масиву N:"); scanf\_s("%d", &count);

printf("Введіть початкове значення масиву a (ціле):"); scanf\_s("%d", &a);

printf("Введіть кінцеве значення масиву b (ціле):"); scanf\_s("%d", &b);

if (a == b) {

printf("Помилка. Значення а не > b.");

}

else {

printf("Масив N(до впорядкування за зростанням(крім 1 елемента):\n");

// Побудова масиву N

for (i = 0; i < count; i++) {

arrN[i] = a + rand() % (b - a + 1);

printf(" %d ", arrN[i]);

}

// Побудова масиву за зростанням залишивши 1 елемент масиву на своєму місці.

do {

flag = 0;

for (i = 1; i < count - 1; i++) {

if (arrN[i] > arrN[i + 1])

{

temp = arrN[i];

arrN[i] = arrN[i + 1];

arrN[i + 1] = temp;

flag = 1;

}

}

} while (flag);

printf("\nУпорядкований масив за зростанням залишивши 1 елемент масиву на своєму місці:\n");

for (i = 0; i < count; i++) {

printf(" %d ", arrN[i]);

}

// Побудова повного масиву за зростанням з підстановкою 1 елемента в масив у потрібне місце.

do {

flag = 0;

for (i = 0; i < count - 1; i++) {

if (arrN[i] > arrN[i + 1])

{

temp = arrN[i];

arrN[i] = arrN[i + 1];

arrN[i + 1] = temp;

flag = 1;

}

}

} while (flag);

printf("\nУпорядкований масив за зростанням:\n");

for (i = 0; i < count; i++) {

printf(" %d ", arrN[i]);

}

}

return(0);

}

3.1:

#include <iostream>

#include <math.h>

#include <windows.h>

#include <time.h>

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(0));

int arrN[1000], count, i, a, b, flag, temp, k, j, last;

printf("Завдання 3(3.1):\nДано масив розміру N і ціле число k (1 <k< N).\nВидалити з масиву елемент із порядковим номером k.\n");

printf("Введіть цілий розмір масиву N:"); scanf\_s("%d", &count);

printf("Введіть початкове значення масиву a (ціле):"); scanf\_s("%d", &a);

printf("Введіть кінцеве значення масиву b (ціле):"); scanf\_s("%d", &b);

printf("Умова: (1 <k< N)\nВведіть ціле число k(індекс), яке задає який елемент масиву видалити:"); scanf\_s("%d", &k);

if (k < 1 && k < count) {

printf("Помилка. Значення k не задовільняє умові: (1 <k< N)");

}

if (a == b) {

printf("Помилка. Значення а не > b.");

}

else {

printf("Масив N(до видалення елемента с індексом ):\n");

// Побудова масиву N

for (i = 0; i < count; i++) {

arrN[i] = a + rand() % (b - a + 1);

printf(" %d ", arrN[i]);

}

// Видалення елементу масива з масива та зрушення елементів у ліво на 1.

arrN[k] = 0;

for (j = 1; j <= 1; j++) {

for (i = k; i <= count - 1; i++) {

arrN[i] = arrN[i + 1];

}

}

// Побудова масиву з видаленим елементом k

printf("\nНовий масив з видаленим k елементом масиву:\n");

for (i = 0; i < count - 1; i++) {

printf(" %d ", arrN[i]);

}

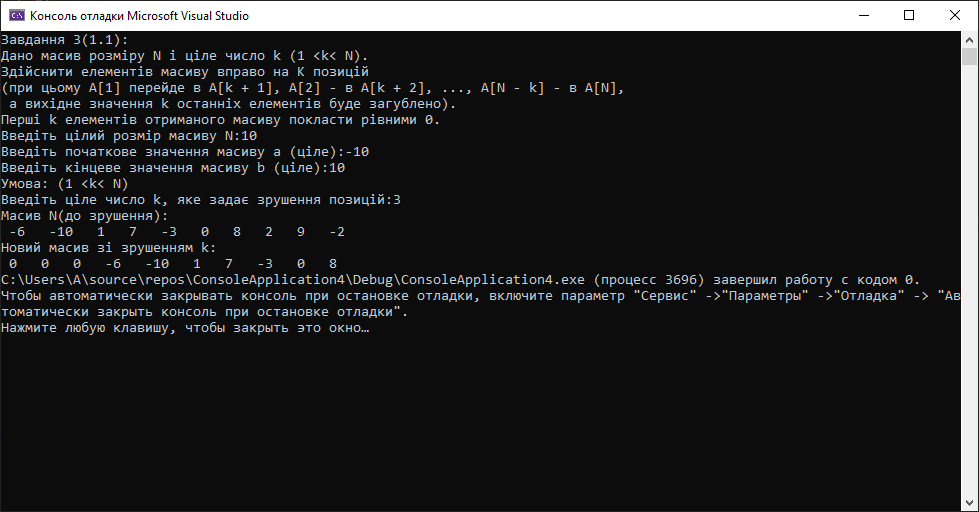
return(0);

}

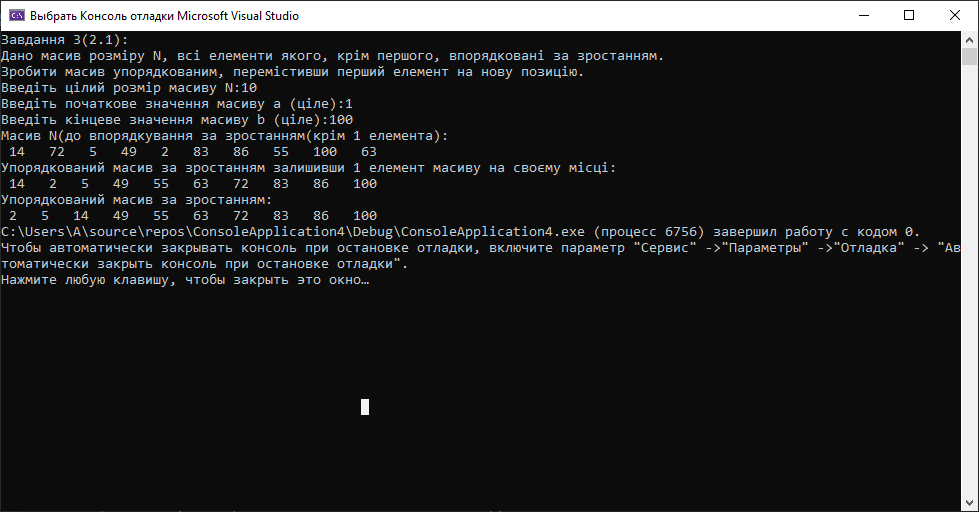
}

Результат програми:

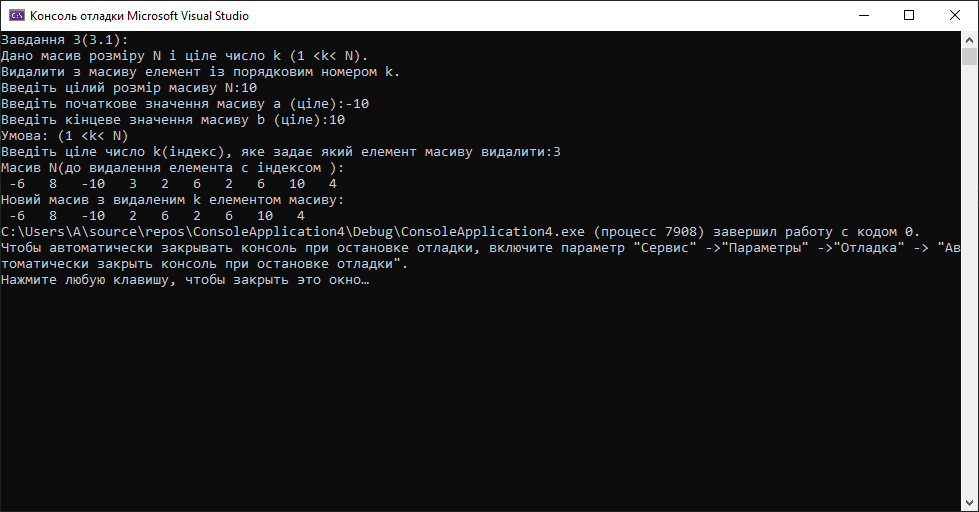
1.1:



2.1:



3.1:



***Висновки:*** в ході виконання лабораторної роботи було ознайомлено з середовищем MS Visual Studio. Досліджено та отримано практичні навики щодо створення найпростішої програми.