Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний

інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 7 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження лінійного пошуку в послідовностях»

Варіант 31

Виконав студент ІП-11 Трикош Іван Володимирович

Перевірила Мартинова О. П.

Київ 2021

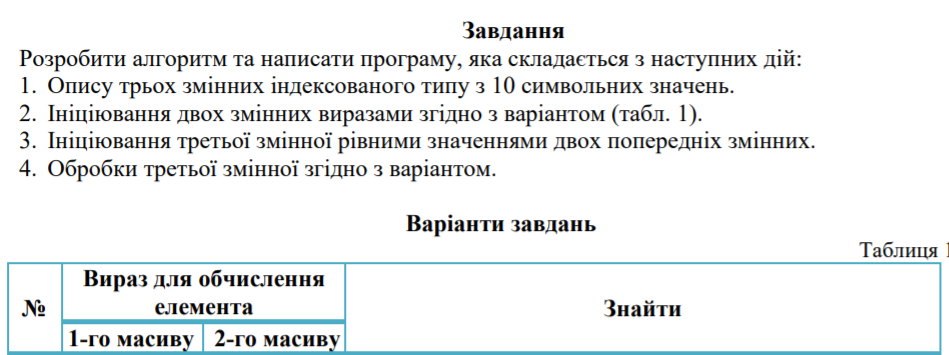
**Лабораторна робота 7**

**Дослідження лінійного пошуку в послідовностях**

**Мета –** дослідити методи послідовного пошуку у впорядкованих і

невпорядкованих послідовностях та набути практичних навичок їх використання під час складання програмних специфікацій.

Варіант 31.





**Постановка задачі –** спочатку ініціюємо три масиви. Кожному елементу першого масиву присвоїти 120 – і (номер елемента); кожному елементу другого масиву присвоїти 110 + і (номер елемента); і якщо елемент першого масиву = елементу другого масиву, то присвоїти це значення елементу третього масиву. Потім у третьому масиві знаходимо максимальний і середньоарифметичний елемент й замінюємо максимальний елемент на середньоарифметичний. Виводимо оброблений масив.

**Побудова математичної моделі**

Складемо таблицю імен змінних:

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Кількість елементів масиву | Цілий (const) | n | Початкове дане |
| Кількість елементів масиву | Цілий | m | Початкове дане |
| Перший масив | Символьний | array1 | Початкове дане |
| Другий масив | Символьний | array2 | Початкове дане |
| Третій масив | Символьний | array3 | Початкове дане і результат |
| Параметр циклу | Цілий | i | Лічильник |
| Параметр циклу | Цілий | j | Лічильник |
| Параметр циклу | Цілий | k | Лічильник |
| Середнє арифм. елем. масиву | Цілий | a | Проміжне дане |
| Індекс максимального елем. | Цілий | b | Проміжне дане |
| Максимальний елем. масиву | Символьний | max | Проміжне дане |
| Сума елементів масиву | Цілий | sum | Проміжне дане |
| Покажчик на array1 | Вказує на символьний тип | p | Покажчик |
| Покажчик на array2 | Вказує на символьний тип | g | Покажчик |
| Покажчик на array3 | Вказує на символьний тип | s | Покажчик |
| Покажчик на m | Вказує на цілий тип | m | Покажчик |

Таким чином, математичне формулювання задачі зводиться до ініціалізації масивів так, як написано у постановці задачі та заміни максимального елемента третього масиву на середньоарифметичний.

**Псевдокод алгоритму**

*Крок 1. Визначимо основні дії.*

*Крок 2. Оголошуємо масиви та їхній розмір.*

*Крок 3. Деталізуємо функцію для ініціалізації масивів.*

*Крок 4. Деталізуємо дію ініціалізації масивів.*

*Крок 5. Деталізуємо функцію для виведення масивів.*

*Крок 6. Деталізуємо дію виводу масивів.*

*Крок 7. Деталізуємо функцію для обробки третього масиву.*

*Крок 8. Деталізуємо дію обробки третього масиву.*

*Крок 9. Деталізуємо функцію для виводу результату.*

*Крок 10. Виводимо результат.*

**Крок 1**

**Початок**

Оголошуємо масиви та їхній розмір

Викликаємо функцію для ініціалізації масивів

Викликаємо функцію для виводу масивів

Викликаємо функцію для обробки третього масиву

Викликаємо функцію для виводу обробленого масиву

**Кінець**

**Крок 2**

**Початок**

const int n = 10

int m = n

char array1[n]

char array2[n]

char array3[n]

Викликаємо функцію для ініціалізації масивів

Викликаємо функцію для виводу масивів

Викликаємо функцію для обробки третього масиву

Викликаємо функцію для виводу обробленого масиву

**Кінець**

**Крок 3**

**Початок**

const int n = 10

int m = n

char array1[n]

char array2[n]

char array3[n]

Input\_array(array1, array2, array3, &m)

Викликаємо функцію для виводу масивів

Викликаємо функцію для обробки третього масиву

Викликаємо функцію для виводу обробленого масиву

**Кінець**

**Підпрограма**

Input\_array(char \*p, char \*g, char \*s, int \*m)

Ініціалізуємо масиви

**Кінець**

**Крок 4**

**Початок**

const int n = 10

int m = n

char array1[n]

char array2[n]

char array3[n]

Input\_array(array1, array2, array3, &m)

Викликаємо функцію для виводу масивів

Викликаємо функцію для обробки третього масиву

Викликаємо функцію для виводу обробленого масиву

**Кінець**

**Підпрограма**

Input\_array(char \*p, char \*g, char \*s, int \*m)

**повторити** \*m **разів**

\*p = 120 - i

\*g = 110 + i

p++

g++

**все повторити**

**повторити** \*m **разів**

**повторити** \*m **разів**

**якщо** \*(p - \*m + i) == \*(g - \*m + j) **то**

\*(s+i) = \*(g - \*m + j)

**все якщо**

**все повторити**

**все повторити**

**Кінець**

**Крок 5**

**Початок**

const int n = 10

int m = n

char array1[n]

char array2[n]

char array3[n]

Input\_array(array1, array2, array3, &m)

Output\_array(array1, array2, array3, &m)

Викликаємо функцію для обробки третього масиву

Викликаємо функцію для виводу обробленого масиву

**Кінець**

**Підпрограма 1**

Input\_array(char \*p, char \*g, char \*s, int \*m)

**повторити** \*m **разів**

\*p = 120 - i

\*g = 110 + i

p++

g++

**все повторити**

**повторити** \*m **разів**

**повторити** \*m **разів**

**якщо** \*(p - \*m + i) == \*(g - \*m + j) **то**

\*(s+i) = \*(g - \*m + j)

**все якщо**

**все повторити**

**все повторити**

**Кінець**

**Підпрограма 2**

Output\_array(array1, array2, array3, &m)

Виводимо масиви

**Кінець**

**Крок 6**

**Початок**

const int n = 10

int m = n

char array1[n]

char array2[n]

char array3[n]

Input\_array(array1, array2, array3, &m)

Output\_array(array1, array2, array3, &m)

Викликаємо функцію для обробки третього масиву

Викликаємо функцію для виводу обробленого масиву

**Кінець**

**Підпрограма 1**

Input\_array(char \*p, char \*g, char \*s, int \*m)

**повторити** \*m **разів**

\*p = 120 - i

\*g = 110 + i

p++

g++

**все повторити**

**повторити** \*m **разів**

**повторити** \*m **разів**

**якщо** \*(p - \*m + i) == \*(g - \*m + j) **то**

\*(s+i) = \*(g - \*m + j)

**все якщо**

**все повторити**

**все повторити**

**Кінець**

**Підпрограма 2**

Output\_array(char \*p, char \*g, char \*s, int \*m)

**повторити** \*m **разів**

Вивід \*(p + i)

**все повторити**

**повторити** \*m **разів**

Вивід \*(g + i)

**все повторити**

**повторити** \*m **разів**

**якщо** \*(s+i) == 0

\*m = \*m – 1

**повторити** \*m **разів**

\*(s + k) = \*(s + k + 1)

**все повторити**

**все якщо**

Вивід \*(s + i)

**все повторити**

**Кінець**

**Крок 7**

**Початок**

const int n = 10

int m = n

char array1[n]

char array2[n]

char array3[n]

Input\_array(array1, array2, array3, &m)

Output\_array(array1, array2, array3, &m)

Solution(array3, &m)

Викликаємо функцію для виводу обробленого масиву

**Кінець**

**Підпрограма 1**

Input\_array(char \*p, char \*g, char \*s, int \*m)

**повторити** \*m **разів**

\*p = 120 - i

\*g = 110 + i

p++

g++

**все повторити**

**повторити** \*m **разів**

**повторити** \*m **разів**

**якщо** \*(p - \*m + i) == \*(g - \*m + j) **то**

\*(s+i) = \*(g - \*m + j)

**все якщо**

**все повторити**

**все повторити**

**Кінець**

**Підпрограма 2**

Output\_array(char \*p, char \*g, char \*s, int \*m)

**повторити** \*m **разів**

Вивід \*(p + i)

**все повторити**

**повторити** \*m **разів**

Вивід \*(g + i)

**все повторити**

**повторити** \*m **разів**

**якщо** \*(s+i) == 0

\*m = \*m – 1

**повторити** \*m **разів**

\*(s + k) = \*(s + k + 1)

**все повторити**

**все якщо**

Вивід \*(s + i)

**все повторити**

**Кінець**

**Підпрограма 3**

Solution(char\* s, int \*m)

Оброблюємо третій масив

**Кінець**

**Крок 8**

**Початок**

const int n = 10

int m = n

char array1[n]

char array2[n]

char array3[n]

Input\_array(array1, array2, array3, &m)

Output\_array(array1, array2, array3, &m)

Solution(array3, &m)

Викликаємо функцію для виводу обробленого масиву

**Кінець**

**Підпрограма 1**

Input\_array(char \*p, char \*g, char \*s, int \*m)

**повторити** \*m **разів**

\*p = 120 - i

\*g = 110 + i

p++

g++

**все повторити**

**повторити** \*m **разів**

**повторити** \*m **разів**

**якщо** \*(p - \*m + i) == \*(g - \*m + j) **то**

\*(s+i) = \*(g - \*m + j)

**все якщо**

**все повторити**

**все повторити**

**Кінець**

**Підпрограма 2**

Output\_array(char \*p, char \*g, char \*s, int \*m)

**повторити** \*m **разів**

Вивід \*(p + i)

**все повторити**

**повторити** \*m **разів**

Вивід \*(g + i)

**все повторити**

**повторити** \*m **разів**

**якщо** \*(s+i) == 0

\*m = \*m – 1

**повторити** \*m **разів**

\*(s + k) = \*(s + k + 1)

**все повторити**

**все якщо**

Вивід \*(s + i)

**все повторити**

**Кінець**

**Підпрограма 3**

Solution(char\* s, int \*m)

int a, b

int sum = 0

char max = 0

**повторити** \*m **разів**

sum += \*(s + i)

**якщо** \*(s + i) > max

max = \*(s + i)

b = i

**все якщо**

**все повторити**

a = sum / \*m

\*(s + b) = a

**Кінець**

**Крок 9**

**Початок**

const int n = 10

int m = n

char array1[n]

char array2[n]

char array3[n]

Input\_array(array1, array2, array3, &m)

Output\_array(array1, array2, array3, &m)

Solution(array3, &m)

Browse(array3, &m)

**Кінець**

**Підпрограма 1**

Input\_array(char \*p, char \*g, char \*s, int \*m)

**повторити** \*m **разів**

\*p = 120 - i

\*g = 110 + i

p++

g++

**все повторити**

**повторити** \*m **разів**

**повторити** \*m **разів**

**якщо** \*(p - \*m + i) == \*(g - \*m + j) **то**

\*(s+i) = \*(g - \*m + j)

**все якщо**

**все повторити**

**все повторити**

**Кінець**

**Підпрограма 2**

Output\_array(char \*p, char \*g, char \*s, int \*m)

**повторити** \*m **разів**

Вивід \*(p + i)

**все повторити**

**повторити** \*m **разів**

Вивід \*(g + i)

**все повторити**

**повторити** \*m **разів**

**якщо** \*(s+i) == 0

\*m = \*m – 1

**повторити** \*m **разів**

\*(s + k) = \*(s + k + 1)

**все повторити**

**все якщо**

Вивід \*(s + i)

**все повторити**

**Кінець**

**Підпрограма 3**

Solution(char\* s, int \*m)

int a, b

int sum = 0

char max = 0

**повторити** \*m **разів**

sum += \*(s + i)

**якщо** \*(s + i) > max

max = \*(s + i)

b = i

**все якщо**

**все повторити**

a = sum / \*m;

\*(s + b) = a

**Кінець**

**Підпрограма 4**

Browse(char\* s, int \*m)

Вивід обробленого масиву

**Кінець**

**Крок 10**

**Початок**

const int n = 10

int m = n

char array1[n]

char array2[n]

char array3[n]

Input\_array(array1, array2, array3, &m)

Output\_array(array1, array2, array3, &m)

Solution(array3, &m)

Browse(array3, &m)

**Кінець**

**Підпрограма 1**

Input\_array(char \*p, char \*g, char \*s, int \*m)

**повторити** \*m **разів**

\*p = 120 - i

\*g = 110 + i

p++

g++

**все повторити**

**повторити** \*m **разів**

**повторити** \*m **разів**

**якщо** \*(p - \*m + i) == \*(g - \*m + j) **то**

\*(s+i) = \*(g - \*m + j)

**все якщо**

**все повторити**

**все повторити**

**Кінець**

**Підпрограма 2**

Output\_array(char \*p, char \*g, char \*s, int \*m)

**повторити** \*m **разів**

Вивід \*(p + i)

**все повторити**

**повторити** \*m **разів**

Вивід \*(g + i)

**все повторити**

**повторити** \*m **разів**

**якщо** \*(s+i) == 0

\*m = \*m – 1

**повторити** \*m **разів**

\*(s + k) = \*(s + k + 1)

**все повторити**

**все якщо**

Вивід \*(s + i)

**все повторити**

**Кінець**

**Підпрограма 3**

Solution(char\* s, int \*m)

int a, b

int sum = 0

char max = 0

**повторити** \*m **разів**

sum += \*(s + i)

**якщо** \*(s + i) > max

max = \*(s + i)

b = i

**все якщо**

**все повторити**

a = sum / \*m;

\*(s + b) = a

**Кінець**

**Підпрограма 4**

Browse(char\* s, int \*m)

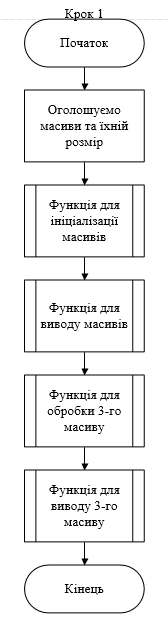
**повторити** \*m **разів**

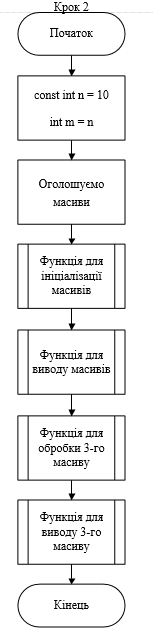
Вивід \*(s+i)

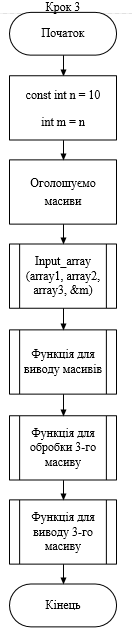
**все повторити**

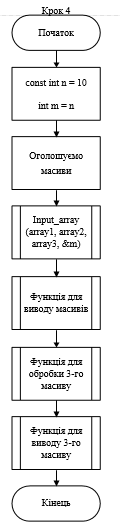
**Кінець**

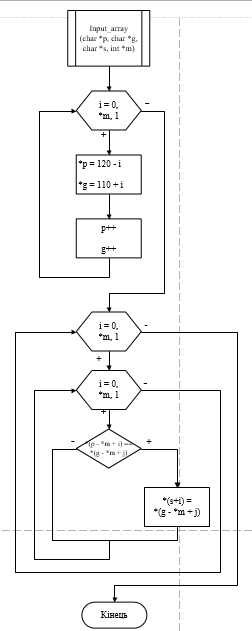
**Блок-схема алгоритму**

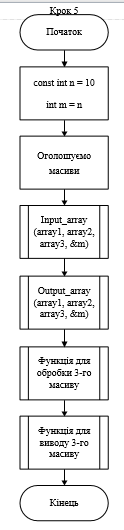


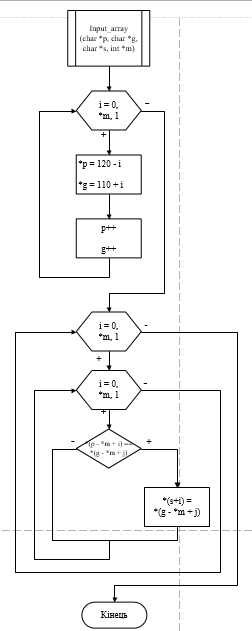


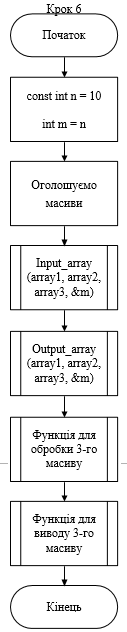


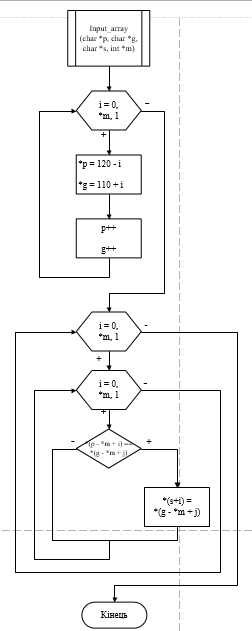


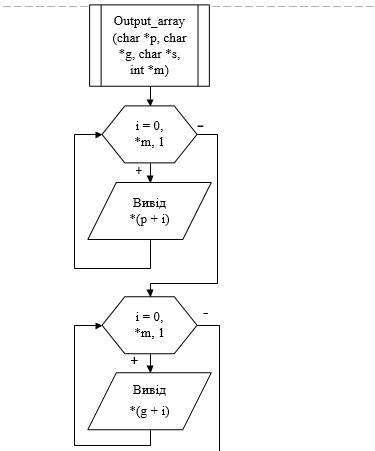


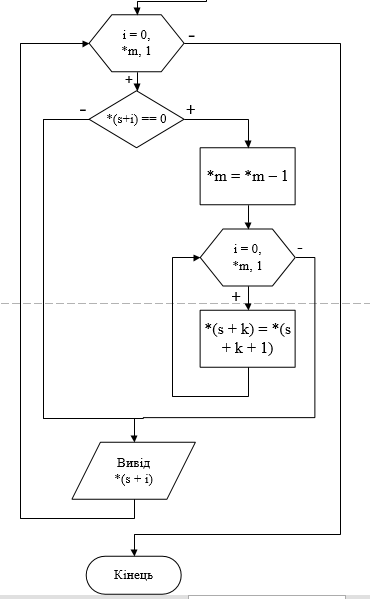


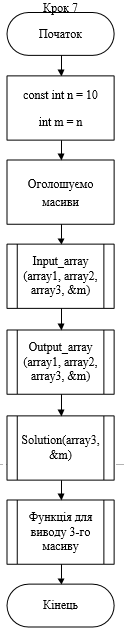


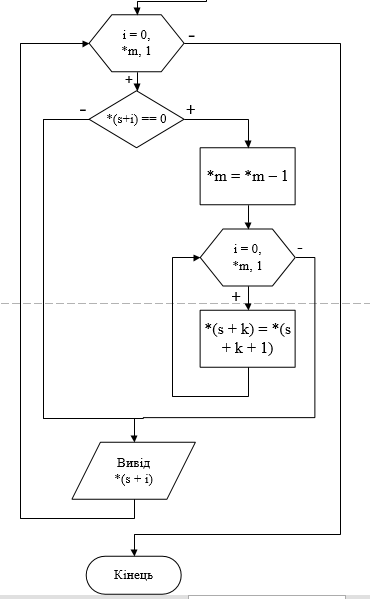
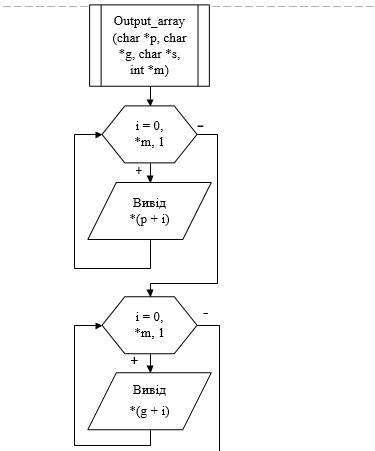


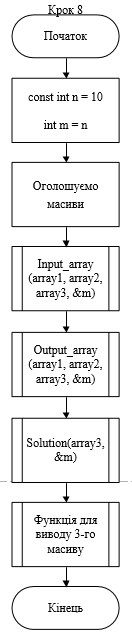


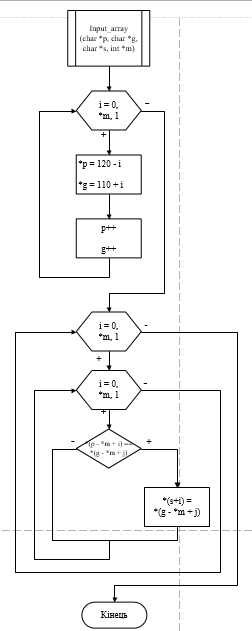


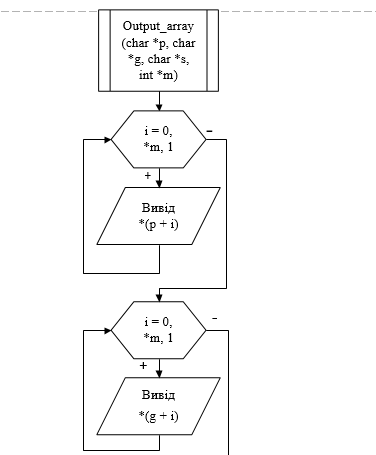


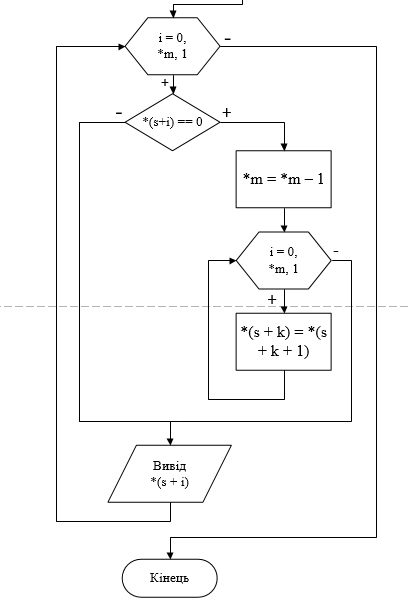


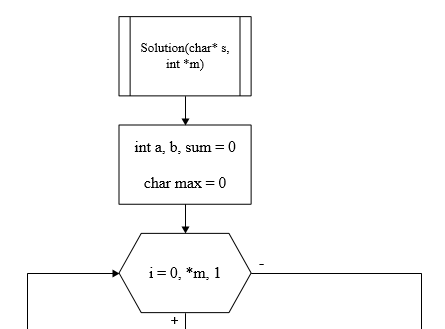


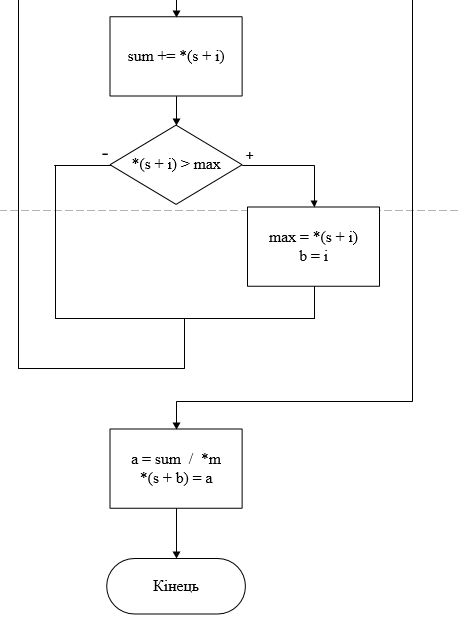
****

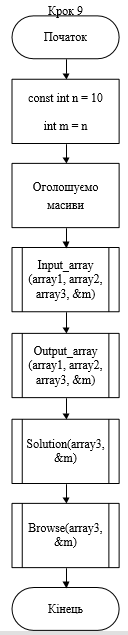


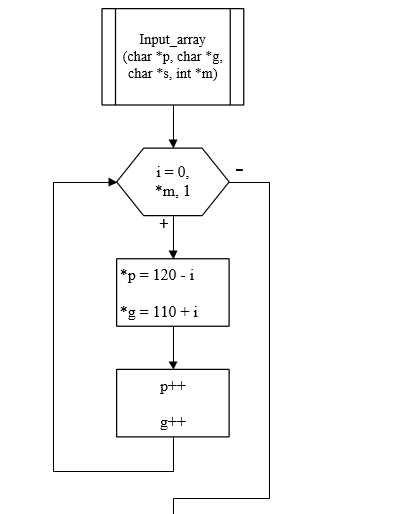
****

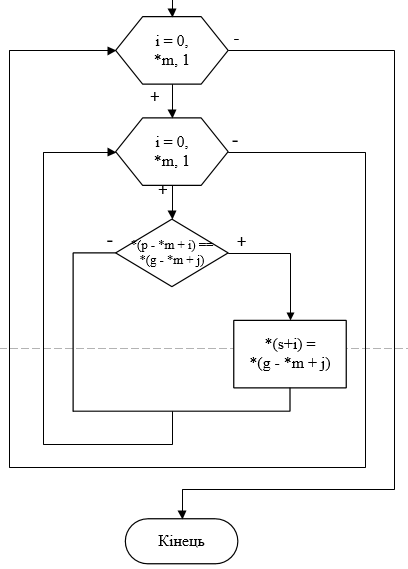
****

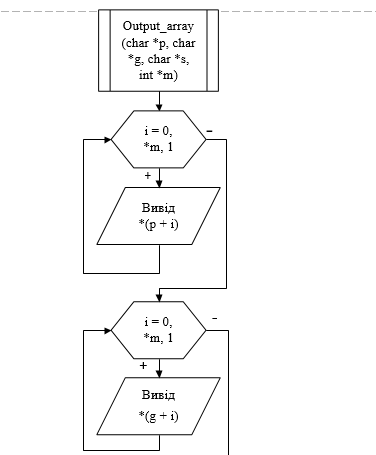
****

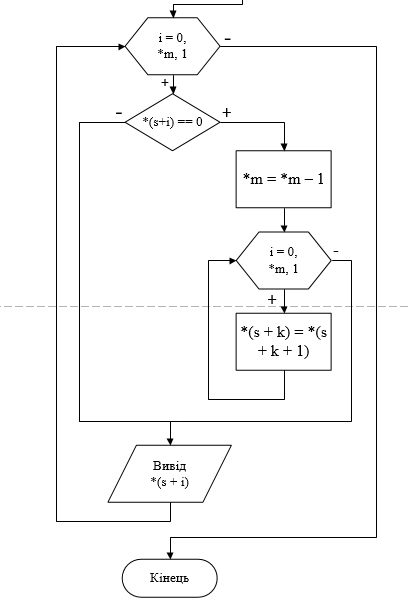
****

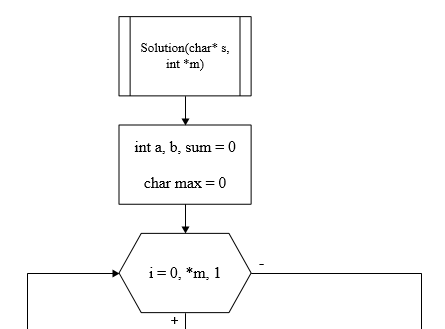
****

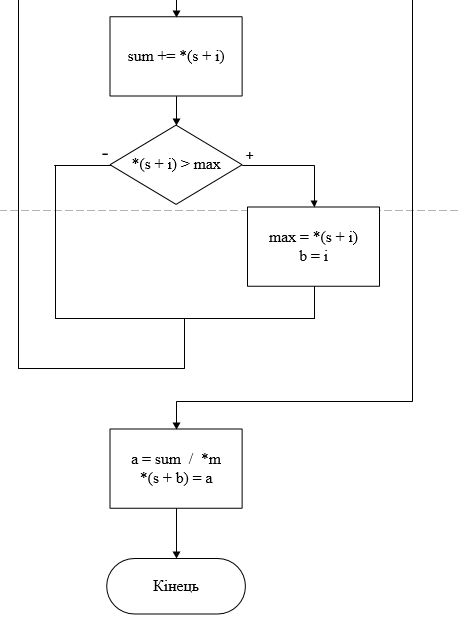
****

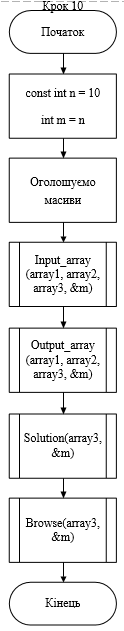
****

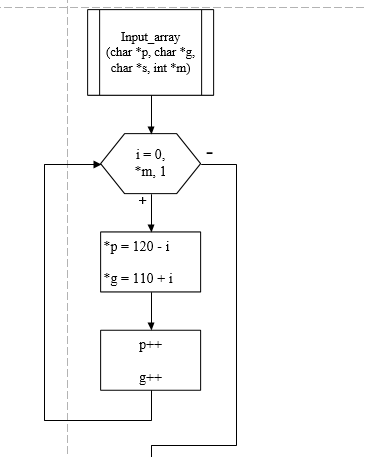
****

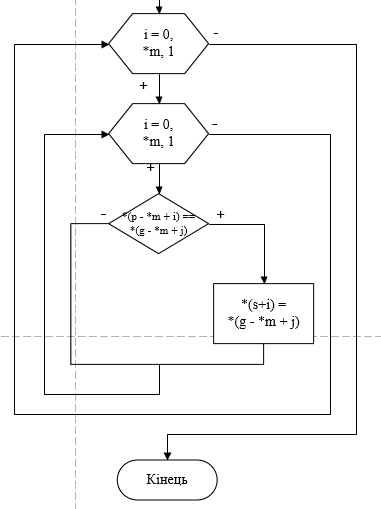
****

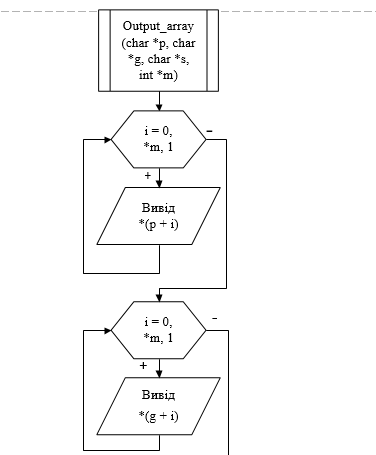
****

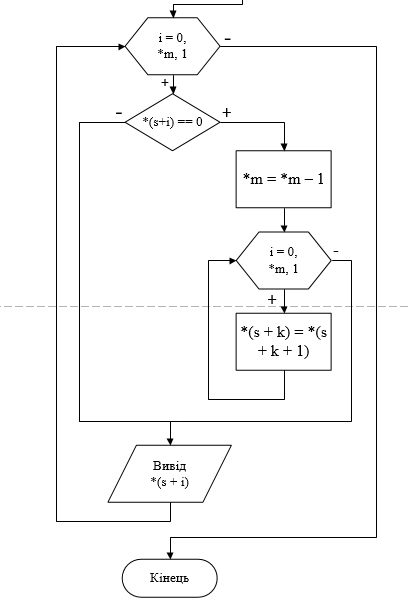
****

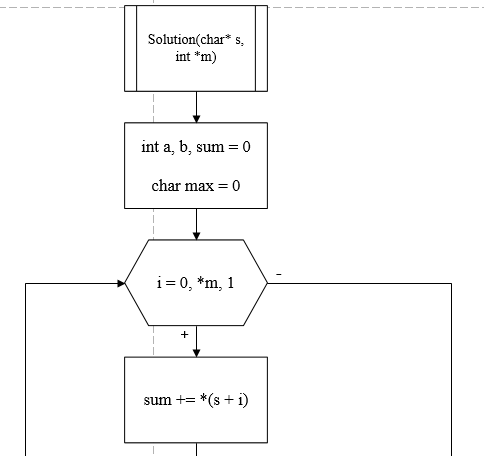
****

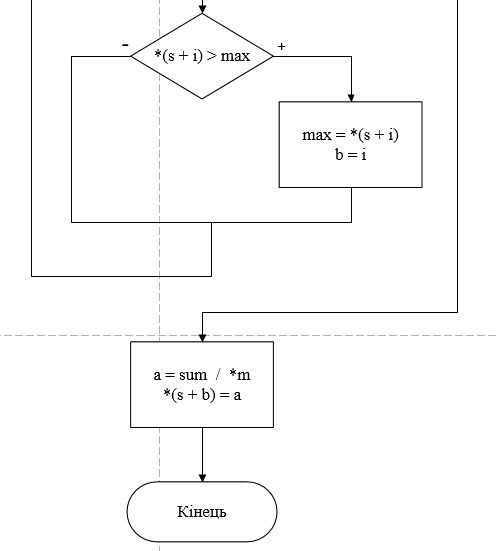
****

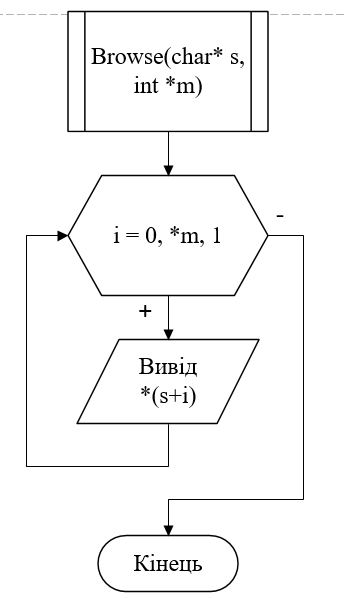
****

****

****

****

****

****

**Код програми на C++:**

#include <iostream>

using namespace std;

void Input\_array(char\*, char\*, char\*, int\*); // Заповнюємо масиви

void Output\_array(char\*, char\*, char\*, int\*); // Виводимо масиви

void Solution(char\*, int\*); // Перетворюємо третій масив

void Browse(char\*, int\*); // Виводимо кінцевий масив

int main()

{

const int n = 10; // Розмір масиву

int m = n; // Змінна, що дорівнює розміру масиву

char array1[m]; // Перший масив

char array2[m]; // Другий масив

char array3[m]; // Третій масив

Input\_array(array1, array2, array3, &m);

Output\_array(array1, array2, array3, &m);

Solution(array3, &m);

Browse(array3, &m);

return 0;

}

void Input\_array(char \*p, char \*g, char \*s, int \*m)

{

// Ініціалізуємо перший і другий масиви

for (int i = 0; i < \*m; i++)

{

\*p = 120 - i;

\*g = 110 + i;

p++;

g++;

}

// Ініціалізуємо третій масив

for (int i = 0; i < \*m; i++)

{

for (int j = 0; j < \*m; j++)

{

if (\*(p - \*m + i) == \*(g - \*m + j))

{

\*(s+i) = \*(g - \*m + j);

}

}

}

}

void Output\_array(char \*p, char \*g, char \*s, int \*m)

{

// Виводимо масиви

cout << "Array1: ";

for (int i = 0; i < \*m; i++)

{

cout << \*(p + i) << " ";

}

cout << "\nArray2: ";

for (int i = 0; i < \*m; i++)

{

cout << \*(g + i) << " ";

}

cout << "\nArray3: ";

for (int i = 0; i < \*m; i++)

{

if (\*(s+i) == 0) //Переміщуємо елементи елементи з нульовим значенням

{

\*m = \*m - 1;

for (int k = 0; k < \*m; k++)

{

\*(s + k) = \*(s + k + 1);

}

}

cout << \*(s + i) << " ";

}

}

void Solution(char\* s, int \*m)

{

int a, b; // a - середнє арифметичне масиву, b - індекс максимального // елемента

int sum = 0; // Сума елементів масиву

char max = 0; // Максимальний елемент масиву

for (int i = 0; i < \*m; i++)

{

sum += \*(s + i); // Знаходимо суму елементів

// Знаходимо максимальне значення масиву

if (\*(s + i) > max)

{

max = \*(s + i);

b = i;

}

}

a = sum / \*m; // Середнє арифметичне елементів масиву

\*(s + b) = a; // Замінюємо максимальний елемент середньоарифметичним

}

void Browse(char\* s, int \*m)

{

// Виводимо кінцевий масив

cout << "\nAnswer: ";

for (int i = 0; i < \*m; i++)

{

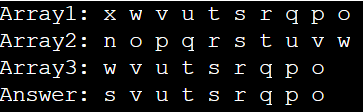
cout << \*(s+i) << " ";

}

}

**Випробування алгоритму**

|  |  |
| --- | --- |
| Блок | Дія |
|  | Початок |
| 1 | Виклик Input\_array(array1, array2, array3, &m) |
|  |  |
|  | Input\_array(char \*p, char \*g, char \*s, int \*m) |
| 2 | array1 = {x, w, v, u, t, s, r, q, p, o} |
| 3 | array2 = {n, o, p ,q, r, s, t, u, v, w} |
|  | Перша ітерація: |
| 4 | array3 = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0} |
|  | Друга ітерація: |
| 5 | array3 = {0, w, 0, 0, 0, 0, 0, 0, 0, 0} |
|  |  |
| 6 | Виклик Output\_array(array1, array2, array3, &m) |
|  |  |
|  | Output\_array(char \*p, char \*g, char \*s, int \*m) |
| 7 | Вивід array1: x, w, v, u, t, s, r, q, p, o |
| 8 | Вивід array2: n, o, p ,q, r, s, t, u, v, w |
|  | Перша ітерація |
|  | \*(s+i) == 0 |
|  | \*m = \*(m - 1) |
|  | k = 0 |
| 9 | array3 = {w, v, u, t, s, r, q, p, 0} |
| 10 | Вивід array3: w, v, u, t, s, r, q, p, o |
|  |  |
| 11 | Виклик Solution(array3, &m) |
|  |  |
|  | Solution(char\* s, int \*m) |
| 12 | sum = 1035 |
| 13 | max = ‘w’ |
| 14 | b = 1 |
| 15 | a = 115 |
| 16 | \*(s+1) = 115 |
|  |  |
| 17 | Виклик Browse(array3, &m) |
|  |  |
|  | Browse(char\* s, int \*m) |
| 18 | Вивід Answer: s, v, u, t, s, r, q, p, o |
|  | Результат: s, v, u, t, s, r, q, p, o |
|  | Кінець |

****

**Висновок –** дослідив методи послідовного пошуку у впорядкованих і

невпорядкованих послідовностях та набув практичних навичок їх використання під час складання програмних специфікацій, удосконалив свої навички розв’язання прикладних задач з програмування та покращив свої знання в програмуванні на мові C++. Навчився працювати з масивами символів.

У результаті виконання програми одержано масив, в якому максимальний елемент замінено на середньоарифметичний.