

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 ІІІ-15 Ткач Владислав Анатолійович

Перевірив: _____

Київ 2021

Лабораторна робота 6

Дослідження лінійного пошуку в послідовностях

Мета - дослідити методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набути практичних навичок їх використання під час складання програмних специфікацій.

Завдання

Розробити алгоритм та написати програму, яка складається з наступних дій:

1. Опису трьох змінних індексованого типу з 10 символних значень.
2. Ініціювання двох змінних виразами згідно з варіантом (табл. 1).
3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.
4. Обробки третьої змінної згідно з варіантом.

31	$120 - i$	$110 + i$	Замінити середньоарифметичним	максимальний елемент
----	-----------	-----------	----------------------------------	-------------------------

Постановка задачі

Результатом розв'язку є маси із заміною (максимальний елемент потрібно замінити середньоарифметичним). Початкові дані: Два масиви елементів, розмір масивів.

Побудова математичної моделі

Складемо таблицю імен змінних:

Змінна	Тип	Ім'я	Призначення
1 масив	Символ	a	Початкове дане
2 масив	Символ	b	Початкове дане
3 масив	Символ	c	Початкове дане
Розмір масиву	ціле	n	Початкове дане
Сума елементів	ціле	sum	Проміжне дане
Максимальний елемент	ціле	max	Проміжне дане
Номер елементу 3 масиву	ціле	num	Проміжне дане
Вивід масиву	Функція	mas_out()	Виввід даних
Визначення максимального	Функція	max_num()	Обрахунок даних та запис
Визначення середнього	функція	arg()	Обрахунок даних

Для визначенн результату зробимо наступні дії:

Ініціалізуємо два масиви за заданою формулою

Визначимо третій масив шляхом перевірки на ідентичність елементів попередніх масивів за допомогою двох циклів(1 вкладений). Для визначення розміру третього масиву використаємо змінну num.

Потім за допомогою функції max_num() визначимо індекс максимального елементу масиву та повернемо його. Одразу присвоюємо максимальному елементу значення середньоарифметичного за допомогою функції arg().

Загалом $c[\text{max_num()}] = \text{arg}()$.

Якщо кількість елементів які співпадають $= 0$ виведемо відповідне повідомлення і завершимо програму.

Розв'язання

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію запису двох масивів

Крок 3. Деталізуємо дію визначення 3-го масиву

Крок 4. Деталізуємо дію перевірки розміру масиву

Крок 5. Деталізуємо дію заміни максимального елементу на середньоарифметичний

Псевдокод алгоритму

Крок 1

Початок

Запис двох масивів

Визначення 3-го масиву

Перевірка розміру 3-го масиву

Заміна максимального на середньоарифметичний

Кінець

Крок 2

Початок

$n = 10$

для $i < n$ повторити

$a[i] = 120 - i$

$b[i] = 110 - i$

все повторити

Визначення 3-го масиву

Перевірка розміру 3-го масиву

Заміна максимального на середньоарифметичний

Кінець

Крок 3

Початок

$n = 10$

для $i < n$ повторити

$a[i] = 120 - i$

$b[i] = 110 - i$

все повторити

$num = 0$

для $i < n$ повторити

для $j < n$ повторити

якщо $a[i] == b[i]$

$c[num] = a[i]$

$num = num + 1$

все якщо

все повторити

все повторити

Перевірка розміру 3-го масиву

Заміна максимального на середньоарифметичний

Кінець

Крок 4

Початок

$n = 10$

для $i < n$ повторити

$a[i] = 120 - i$

$b[i] = 110 - i$

все повторити

$num = 0$

для $i < n$ повторити

для $j < n$ повторити

якщо $a[i] == b[i]$

$c[num] = a[i]$

$num = num + 1$

все якщо

все повторити

все повторити

якщо $num == 0$

вивід “no same element”

кінець

все якщо

Заміна максимального на середньоарифметичний

Кінець

Крок 5

Початок

$n = 10$

для $i < n$ повторити

$a[i] = 120 - i$

$b[i] = 110 - i$

все повторити

$num = 0$

для $i < n$ повторити

для $j < n$ повторити

якщо $a[i] == b[j]$

$c[num] = a[i]$

$num = num + 1$

все якщо

все повторити

все повторити

якщо $num == 0$

вивід “no same element”

кінець

все якщо

$c[\text{max_num}(c, num)] = \text{arg}(c, num)$

$\text{mas_out}(c, num)$

Кінець

Підпрограми:

max_num(mas[], n)

max = 0

для $i < n$ повторити

якщо $mas[i] > mas[max]$

max = i

все якщо

все повторити

повернути max

arg(mas[], n)

sum = 0

для $i < n$ повторити

sum = sum + mas[i]

все повторити

повернути sum/n

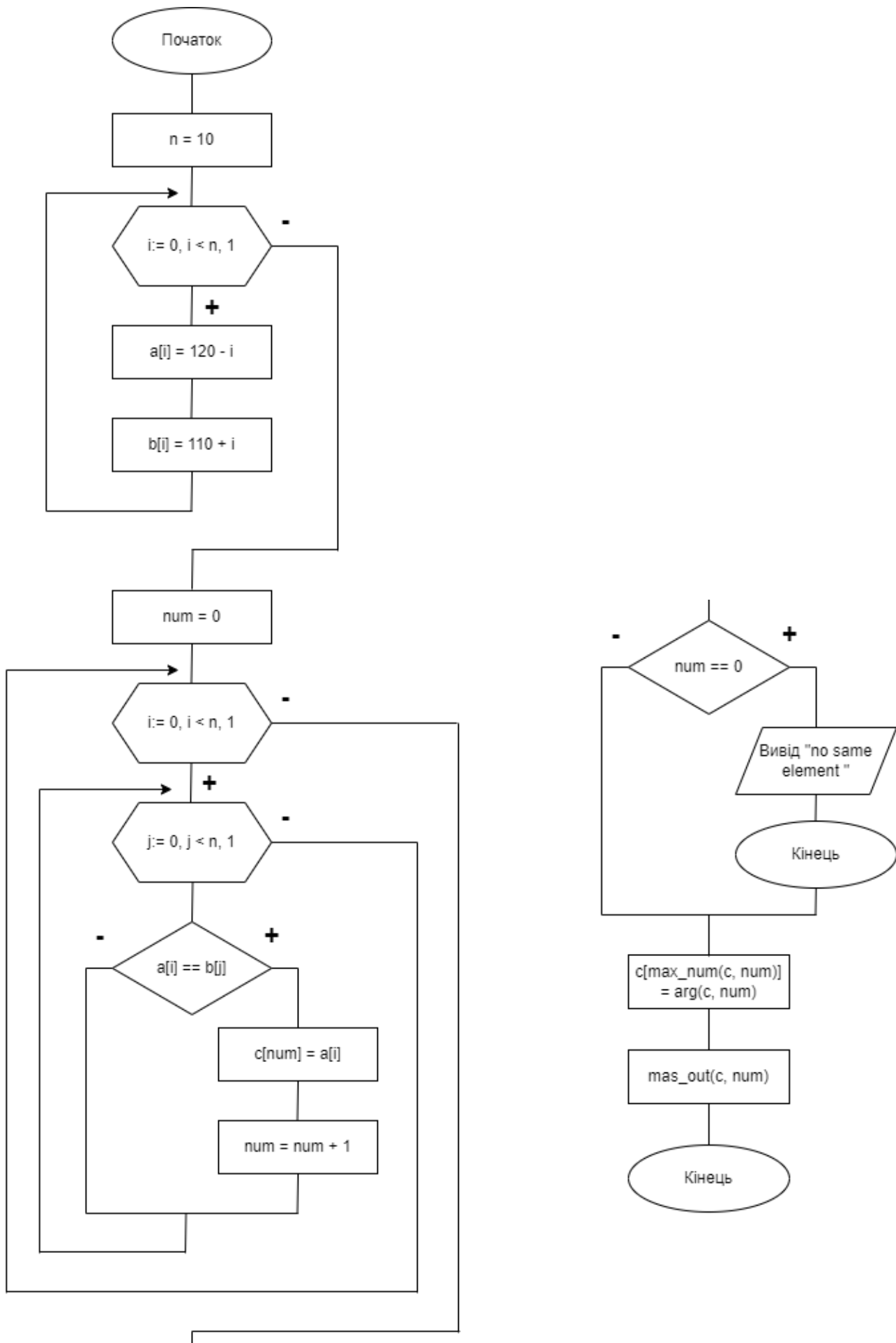
mas_out (mas[], n)

для $i < n$ повторити

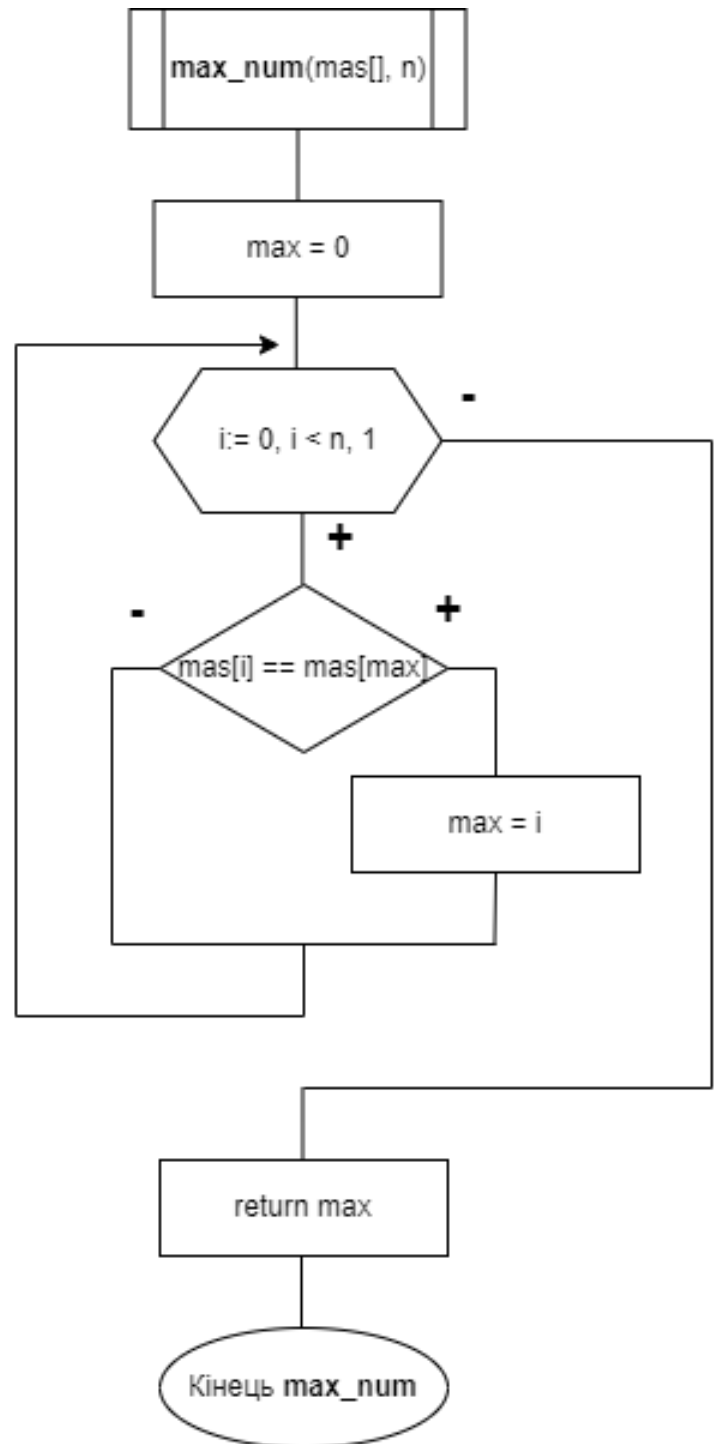
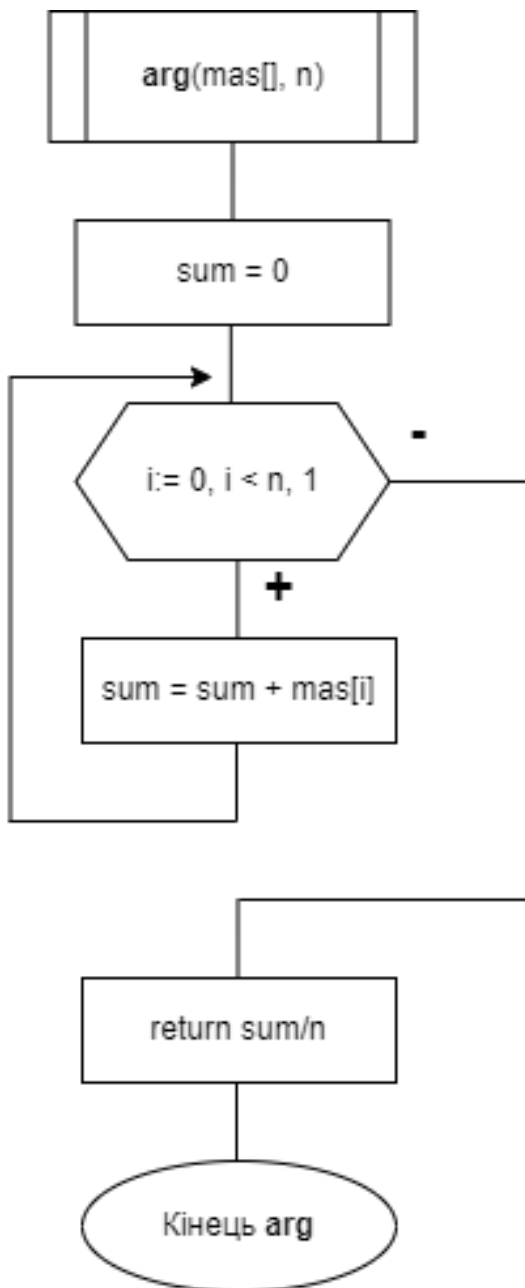
вивід mas[i]

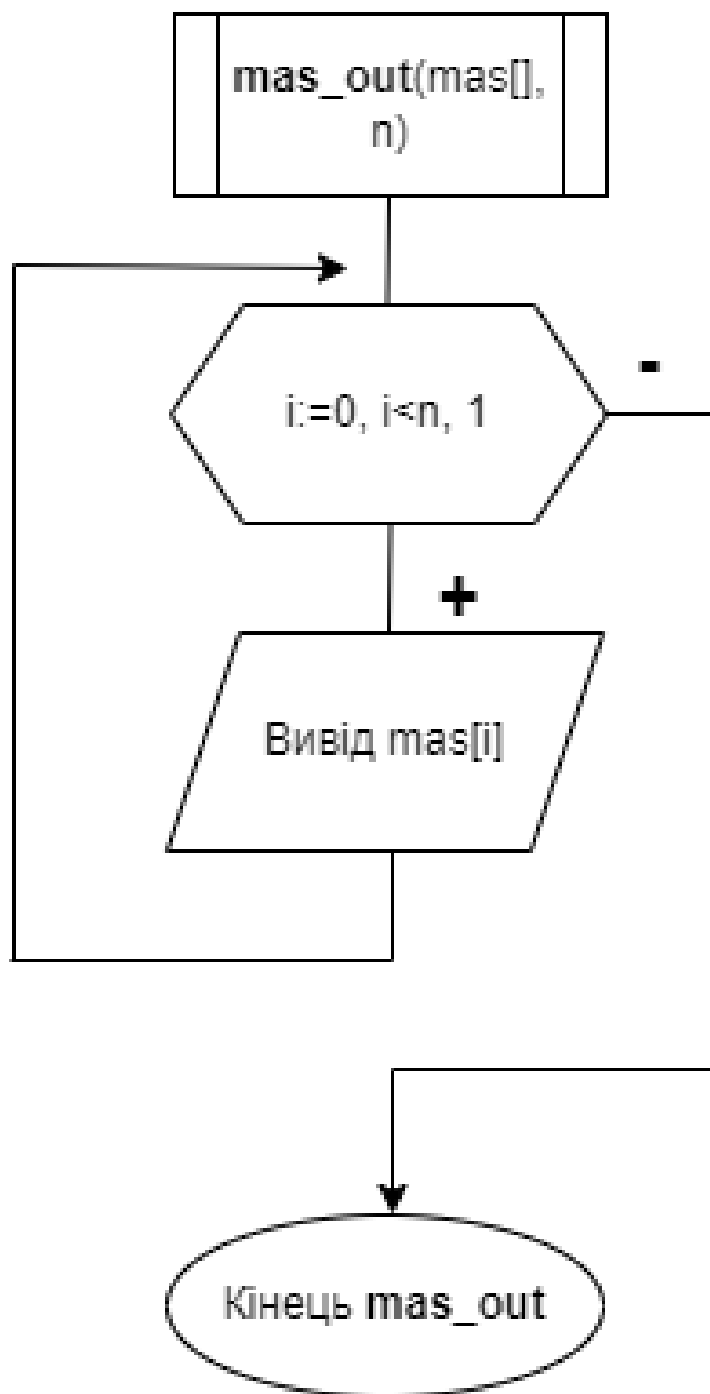
все повторити

Блок схема алгоритму



Підпрограми





Код програми

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int max_num(char *mas, int n){
5      int max = 0;
6      for(int i = 0; i < n; i++){
7          if(mas[i] > mas[max]){
8              max = i;
9          }
10     }
11     return max;
12 }
13
14 double arg(char *mas, int n){
15     int sum = 0;
16     for(int i = 0; i < n; i++){
17         sum += mas[i];
18     }
19     return sum/n;
20 }
21
22 void mas_out(char *mas, int n){
23     for(int i = 0; i < n; i++){
24         cout << mas[i] << " ";
25     }
26     cout << endl;
27 }
28
29 int main(){
30     int n = 10;
31     char a[n], b[n], c[n];
32     for(int i = 0; i < n; i++){
33         a[i] = 120 - i;
34         b[i] = 110 + i;
35     }
36     mas_out(a, n);
37     mas_out(b, n);
38     int num = 0;
39     for(int i = 0; i < n; i++)
40         for(int j = 0; j < n; j++)
41             if(a[i] == b[j]){
42                 c[num] = a[i];
43                 num++;
44             }
45
46     if(num == 0)
47         cout << "no same element";
48
49     c[max_num(c, num)] = arg(c, num);
50     mas_out(c, num);
51 }

```

Випробування програми

```
x w v u t s r q p o
n o p q r s t u v w
s v u t s r q p o

-----
Process exited after 0.07327 seconds with return value 0
Press any key to continue . . . _
```

Висновки

Ми дослідили методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набули практичних навичок їх використання під час складання програмних специфікацій..

У результаті виконання лабораторної роботи ми отримали алгоритм для заміни максимального елементу масиву на середньоарифметичне: визначили основні дії, деталізували дію запису двох масивів, дію визначення 3-го масиву, дію перевірки розміру 3-го масиву та дію заміни максимального елементу на середньоарифметичний.