

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 12

Виконала студентка ІП-15 Коваленко Марія Олександрівна
(шифр, прізвище, ім'я, по батькові)
Перевірила Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Лабораторна робота 7

Дослідження лінійного пошуку в послідовностях

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

Індивідуальне завдання

Варіант 12

Завдання

12.

Постановка задачі

Знайти суму елементів, коди яких більше 101

Побудова математичної моделі

Складемо таблицю імен змінних.

Змінна	Тип	Ім'я	Призначення
лічильник	цілий	i	проміжні дані
лічильник	цілий	k	проміжні дані
відповідь	цілий	answer	вихідні дані
масив у функції	символьний	array	проміжні дані
масив у функції	символьний	array_1	проміжні дані
масив у функції	символьний	array_2	проміжні дані

sum_array – функція для пошуку суми

array_3 — функція для пошуку третього масиву

array_1 - функція для пошуку першого масиву

array_2 - функція для пошуку другого масиву

Розв'язання

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

Крок 1. Визначимо основні дії;

Крок 2. Деталізуємо знаходження суми необхідних елементів

Псевдокод

початок

answer:=sum_array(array_3(array_1(), array_2()))

виведення answer

кінець

підпрограми

array_1()

повторити для i від 0 до 10

array[i]:=95 + i

кінець циклу

return array

кінець

array_2()

повторити для i від 0 до 10

array[i]:=100-i

кінець циклу
return array

кінець

array_3(array_1, array_2)

повторити для i від 0 до 10

array[i]=0

повторити для k від 1 до 10

якщо array_1[i]= array_2[k]

то array[i]:=array[i]+array_1[i]

кінець циклу

кінець циклу

return array

кінець

sum_array(array)

sum:=0

повторити для i від 0 до 10

якщо array[i]>101

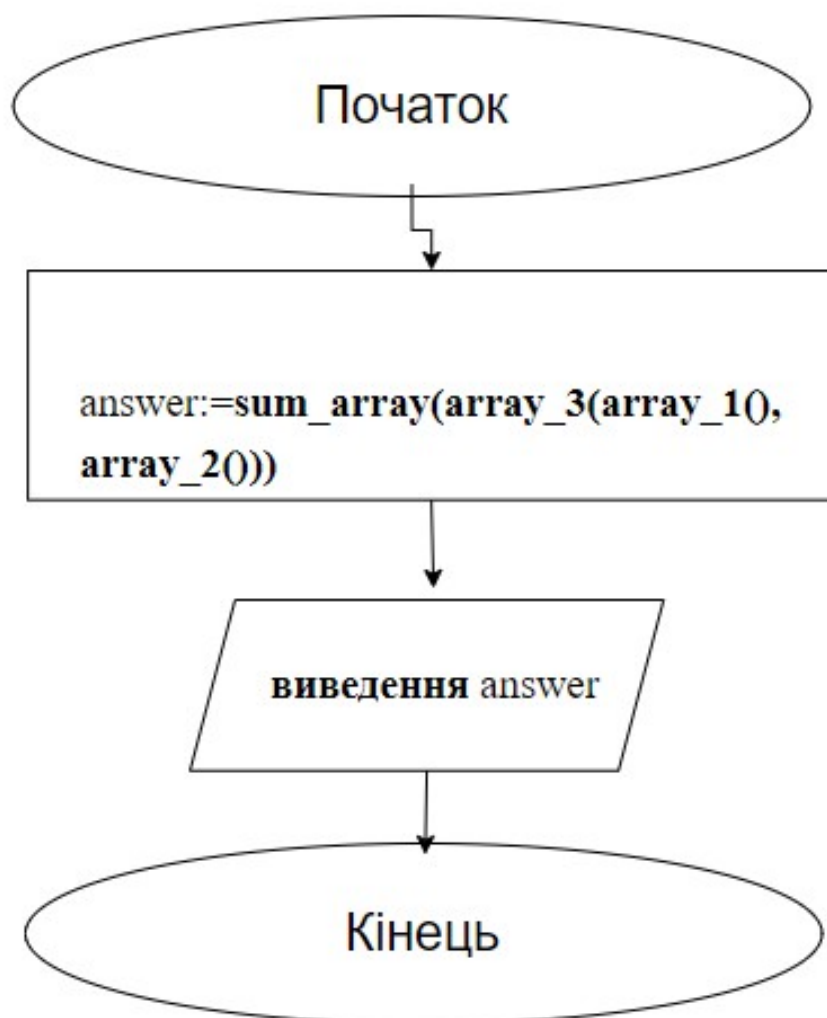
то sum:=sum+array[i]

кінець циклу

return sum

кінець

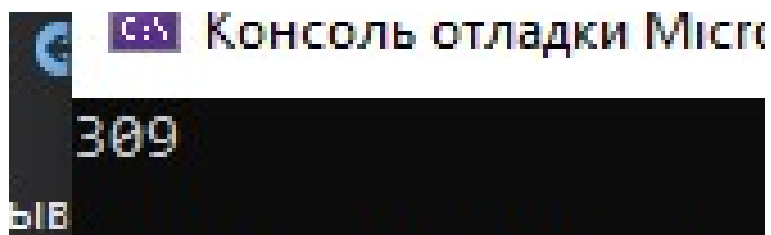
Блок-схеми



Код

```
lab_7
4 {
5     Ссылка: 0
6     class Program
7     {
8         Ссылка: 0
9         static void Main(string[] args)
10        {
11            int answer = sum(array_3(array_1(), array_2()));
12            Console.WriteLine(answer);
13        }
14        Ссылка: 1
15        static char[] array_1()
16        {
17            char[] array = new char [10];
18            for (int i = 0; i < 10; i++)
19            {
20                array[i] = Convert.ToChar( i + 95 );
21            }
22            return array;
23        }
24        Ссылка: 1
25        static char[] array_2()
26        {
27            char[] array = new char[10];
28            for (int i = 0; i < 10; i++)
29            {
30                array[i] = Convert.ToChar(105 - i);
31            }
32            return array;
33        }
34        Ссылка: 1
35        static char[] array_3(char[] array_1, char[] array_2)
36        {
37            char[] array = new char[10];
38            for (int i = 0; i < 10; i++)
39            {
40                array[i] = Convert.ToChar(0);
41                for (int k = 0; k < 10; k++)
42                {
43                    if (array_1[i] == array_2[k]){
44                        array[i] = Convert.ToChar(array[i] + array_1[i]);
45                    }
46                }
47            }
48            return array;
49        }
50        Ссылка: 1
51        static int sum(char[] array)
52        {
53            int sum = 0;
54            for (int i = 0; i < 10; i++)
55            {
56                if (Convert.ToInt32(array[i]) > 101)
57                {
58                    sum = sum + Convert.ToInt32(array[i]);
59                }
60            }
61            return sum;
62        }
63    }
64 }
```

Тестування



Висновок:

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