Міністерство освіти і науки України

Національний технічний університет України “Київський політехнічний

інститут імені Ігоря Сікорського ˮ

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 7 з дисципліни

“Алгоритми та структури даних-1.

Основи алгоритмізації ˮ

“ Дослідження лінійних алгоритмів ˮ

Варіант:12

Виконав студент: ІП-12 Єльчанінов Артем Юрійович

(шифр, прізвище, ім'я, по батькові)

Перевірив: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище, ім'я, по батькові)

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 12**

**Задача:** Знайти суму елементів, коди яких більше 101

Вираз для обчислення 1-го масиву: 95 + і

Вираз для обчислення 2-го масиву: 105 – і

**Постановка задачі**

Результатом розв’язку задачі є знаходження суми елементів третього масиву, коди яких більше 101.

Спершу заповнюємо за даними умовою задачі виразами для знаходження елементів перший та другий масиви, але враховуючи специфіку задачі ми ще в дії заповнення масивів заповнюємо третій масив пустими елементами. Потім виводимо на екран за допомогою допоміжної функції елементи 1-го та 2-го масивів. Далі відбувається заповнення 3-го масиву рівними елементами 1-го та 2-го масивів, після цієї дії виведення на екран елементів масиву. Останньою дією є знаходження суми елементів, коди яких більше 101, яке відбувається перебором елементів 3-го масиву, і якщо код елемента більше 101, то він додається до змінної, яка зберігатиме значення суми. І після виконання цієї дії задача буде виконана.

**Математична модель**

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Розмір масивів | Цілий | size | Вхідне дане |
| Перший масив | Символьний | array\_1 | Промідне дане |
| Другий масив | Символьний | array\_2 | Проміжне дане |
| Третій масив | Символьний | array\_3 | Проміжне дане |
| Лічильник для арифметичних цилів | Цілий | i | Проміжне дане |
| Додатковий лічильник для арифметичного циклу | Цілий | j | Проміжне дане |
| Сума елементів, коди яких більше 101 | Цілий | sum | Вихідне дане |
| Функція для виведення значень масиву | Відсутній  (void) | output\_array | Допоміжний алгоритм |

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

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію заповнення трьох масивів.

Крок 3. Деталізуємо дію заповнення 3-го масиву рівними елементами 1-го та 2-го масивів.

Крок 4. Деталізуємо дію обчислення суми.

**Псевдокод алгоритму**

**Крок 1:**

**Початок**

Ініціалізація змінних

Заповнення трьох масивів

Виведення елементів першого масиву

Виведення елементів другого масиву

Перебір елементів 1-го масиву

Перебір елементів 2-го масиву

Заповнення 3-го масиву рівними елементами 1-го та 2-го масивів

Виведення елементів третього масиву

Обчислення суми

**Виведення** sum

**Кінець**

**Крок 2:**

**Початок**

size:= 10; sum:= 0;

Заповнення трьох масивів

Виведення елементів першого масиву

Виведення елементів другого масиву

Перебір елементів 1-го масиву

Перебір елементів 2-го масиву

Заповнення 3-го масиву рівними елементами 1-го та 2-го масивів

Виведення елементів третього масиву

Обчислення суми

**Виведення** sum

**Кінець**

**Крок 3:**

**Початок**

size:= 10; sum:= 0;

**для** і **від** 0 **до** size **з кроком** 1

array\_1[i] := 95 + i

array\_2[i] := 105 – i

array\_3[i] : = 0

**все повторити**

Виведення елементів першого масиву

Виведення елементів другого масиву

Перебір елементів 1-го масиву

Перебір елементів 2-го масиву

Заповнення 3-го масиву рівними елементами 1-го та 2-го масивів

Виведення елементів третього масиву

Обчислення суми

**Виведення** sum

**Кінець**

**Крок 4:**

**Початок**

size:= 10; sum:= 0;

**для** і **від** 0 **до** size **з кроком** 1

array\_1[i] := 95 + i

array\_2[i] := 105 – i

array\_3[i] : = 0

**все повторити**

output\_array(array\_1, size)

Виведення елементів другого масиву

Перебір елементів 1-го масиву

Перебір елементів 2-го масиву

Заповнення 3-го масиву рівними елементами 1-го та 2-го масивів

Виведення елементів третього масиву

Обчислення суми

**Виведення** sum

**Кінець**

**Крок 5:**

**Початок**

size:= 10; sum:= 0;

**для** і **від** 0 **до** size **з кроком** 1

array\_1[i] := 95 + i

array\_2[i] := 105 – i

array\_3[i] : = 0

**все повторити**

output\_array(array\_1, size)

output\_array(array\_2, size)

Перебір елементів 1-го масиву

Перебір елементів 2-го масиву

Заповнення 3-го масиву рівними елементами 1-го та 2-го масивів

Виведення елементів третього масиву

Обчислення суми

**Виведення** sum

**Кінець**

**Крок 6:**

**Початок**

size:= 10; sum:= 0;

**для** і **від** 0 **до** size **з кроком** 1

array\_1[i] := 95 + i

array\_2[i] := 105 – i

array\_3[i] : = 0

**все повторити**

output\_array(array\_1, size)

output\_array(array\_2, size)

**для** і **від** 0 **до** size **з кроком** 1

Перебір елементів 2-го масиву

Заповнення 3-го масиву рівними елементами 1-го та 2-го масивів

**все повторити**

Виведення елементів третього масиву

Обчислення суми

**Виведення** sum

**Кінець**

**Крок 7:**

**Початок**

size:= 10; sum:= 0;

**для** і **від** 0 **до** size **з кроком** 1

array\_1[i] := 95 + i

array\_2[i] := 105 – i

array\_3[i] : = 0

**все повторити**

output\_array(array\_1, size)

output\_array(array\_2, size)

**для** і **від** 0 **до** size **з кроком** 1

**для** j **від** 0 **до** size **з кроком** 1

Заповнення 3-го масиву рівними елементами 1-го та 2-го масивів

**все повторити**

**все повторити**

Виведення елементів третього масиву

Обчислення суми

**Виведення** sum

**Кінець**

**Крок 8:**

**Початок**

size:= 10; sum:= 0;

**для** і **від** 0 **до** size **з кроком** 1

array\_1[i] := 95 + i

array\_2[i] := 105 – i

array\_3[i] : = 0

**все повторити**

output\_array(array\_1, size)

output\_array(array\_2, size)

**для** і **від** 0 **до** size **з кроком** 1

**для** j **від** 0 **до** size **з кроком** 1

**якщо** array\_1[i] == array\_2[j]

**то**

array\_3[i] := array\_1[i]

**все якщо**

**все повторити**

**все повторити**

Виведення елементів третього масиву

Обчислення суми

**Виведення** sum

**Кінець**

**Крок 9:**

**Початок**

size:= 10; sum:= 0;

**для** і **від** 0 **до** size **з кроком** 1

array\_1[i] := 95 + i

array\_2[i] := 105 – i

array\_3[i] : = 0

**все повторити**

output\_array(array\_1, size)

output\_array(array\_2, size)

**для** і **від** 0 **до** size **з кроком** 1

**для** j **від** 0 **до** size **з кроком** 1

**якщо** array\_1[i] == array\_2[j]

**то**

array\_3[i] := array\_1[i]

**все якщо**

**все повторити**

**все повторити**

output\_array(array\_3, size)

Обчислення суми

**Виведення** sum

**Кінець**

**Крок 10:**

**Початок**

size:= 10; sum:= 0;

**для** і **від** 0 **до** size **з кроком** 1

array\_1[i] := 95 + i

array\_2[i] := 105 – i

array\_3[i] : = 0

**все повторити**

output\_array(array\_1, size)

output\_array(array\_2, size)

**для** і **від** 0 **до** size **з кроком** 1

**для** j **від** 0 **до** size **з кроком** 1

**якщо** array\_1[i] == array\_2[j]

**то**

array\_3[i] := array\_1[i]

**все якщо**

**все повторити**

**все повторити**

output\_array(array\_3, size)

**для** і **від** 0 **до** size **з кроком** 1

**якщо** array\_3[i] > 101

**то**

sum += array\_3[i]

**все якщо**

**все повторити**

**Виведення** sum

**Кінець**

**output\_array(array[], size):**

**Початок**

**для** і **від** 0 **до** size **з кроком** 1

**Виведення** array[i]

**все повторити**

**Кінець**

**Блок-схема**

**Крок 1: Крок 2:**

**Крок 3: Крок 4:**

**Крок 5:**



**Крок 6:**



**Крок 7:**



**Крок 8:**



**Крок 9:**



**Крок 10:**



**output\_array(array[], size):**



**Код програми на мові С++ :**

#include <iostream>

#include <iomanip>

using namespace std;

void output\_array(char array[], int size);

int main() {

int sum = 0;

const int size = 10;

char array\_1[size], array\_2[size], array\_3[size];

for (int i = 0; i < size; i++) {

array\_1[i] = 95 + i;

array\_2[i] = 105 - i;

array\_3[i] = 0;

}

cout << "array\_1:" << endl;

output\_array(array\_1, size);

cout << "array\_2:" << endl;

output\_array(array\_2, size);

for (int i = 0; i < size; i++) {

for (int j = 0; j < size; j++) {

if (array\_1[i] == array\_2[j]) {

array\_3[i] = array\_1[i];

}

}

}

cout << "array\_3:" << endl;

output\_array(array\_3, size);

for (int i = 0; i < size; i++) {

if (array\_3[i] > 101) {

sum += array\_3[i];

}

}

cout << "sum: " << sum << endl;

system("pause");

return 0;

}

void output\_array(char array[], int size) {

for (int i = 0; i < size; i++) {

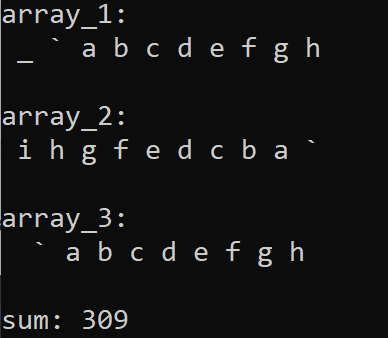
cout << setw(2) << array[i];

}

cout << endl << endl;

}

**Тестування програми:**



**Висновок.**

У результаті лабораторної роботи було розроблено математичну модель, що відповідає постановці задачі; псевдокод та блок-схеми, які пояснюють логіку алгоритму. Було набуто практичного новичок у використанні методів лінійного пошуку в послідовностях та їх інтерпретації у блок-схеми і псевдокод.

Алгоритм був випробуваний з використанням значень виразів для знаходження елементів 1-го( 95 + і ) та 2-го( 105 – i ) масивів, заданих умовою задачі. У підсумку було отримано, що sum = 309. Таким чином, було доведено вірність складеного алгоритму. Отже, його можна застосовувати для лінійного пошуку в послідовностях.