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 18

Виконав студент ІП-12 Кушнір Ганна Вікторівна

(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 18**

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

1. Опису трьох змінних індексованого типу з 10 символьних значень.

2. Ініціювання двох змінних виразами згідно з варіантом.

3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.

4. Обробки третьої змінної згідно з варіантом.

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Вираз для обчислення елемента** | | **Знайти** |
| **1-го масиву** | **2-го масиву** |
| **18** | 60 – 2 \* i | 40 + 3 \* i | Перше входження елементу з кодом 52 |

1. *Постановка задачі.* Початковим даним є кількість значень (n) змінних індексованого типу; ця кількість за умовою дорівнює 10. Результатом розв’язку є індекс (номер) першого елементу третьої змінної індексованого типу (третього масиву) C[n], який має код 52.
2. *Побудова математичної моделі.* Складемо таблицю імен змінних.

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Кількість елементів масиву (послідовності) | Цілий, сталий (константа) | n | Початкове дане |
| Перша змінна індексованого типу (перший масив) | Символьний | A[n] | Допоміжна змінна |
| Друга змінна індексованого типу (другий масив) | Символьний | B[n] | Допоміжна змінна |
| Третя змінна індексованого типу (третій масив) | Символьний | C[n] | Допоміжна змінна |
| Змінна для послідовного створення третього масиву | Цілий | k | Допоміжна змінна |
| Параметр арифметичного циклу | Цілий, послідовний | i | Лічильник |
| Параметр арифметичного циклу | Цілий, послідовний | j | Лічильник |
| Формальний параметр для передачі кількості значень індексованих змінних у функцію | Цілий, сталий (константа) | m | Допоміжна змінна |
| Форм. параметр; одновимірний масив, що передається у функцію | Символьний | arr[] | Допоміжна змінна |
| Змінна для індексації масиву arr[] у функції find\_52() | Цілий | u | Лічильник |
| Умова для ітераційного циклу (виконувати, поки істина) | Логічний | r | Допоміжна змінна |
| Форм. парам. для передачі з функції find\_52() номеру першого елем. масиву з кодом 52 | Цілий | num | Допоміжна змінна |
| Номер першого елементу послідовності C[n] з кодом 52 | Цілий | number | Результат |

Складемо таблицю імен допоміжних алгоритмів (функцій).

|  |  |  |
| --- | --- | --- |
| **Функція** | **Тип результату** | **Ім’я** |
| Виведення одновимірного масиву | – | output() |
| Пошук першого входження у масив елементу з кодом 52 | Цілий | find\_52() |

Таким чином, математичне формулювання задачі зводиться до виконання наступних дій:

1. Присвоєння початкових значень змінним n:=10 та k:=0.
2. Створення двох змінних індексованого типу (одновимірних масивів) A[n] та B[n], які за допомогою арифметичного циклу з параметром i (i приймає значення від 1 до n включно) задаються за формулами: A[i]:=60–2\*i, B[i]:=40+3\*i.
3. Наступним кроком відбувається ініціалізація третьої змінної індексованого типу C[n] рівними значеннями перших двох змінних A[] та B[]; кожне зі значень цих двох змінних порівнюється одне з одним за допомогою вкладених арифметичних циклів з параметрами i та j (i приймає значення від 1 до n включно та відповідає за змінну A[], на кожній з ітерацій змінної i j також приймає значення від 1 до n включно та відповідає за змінну B[]).
4. Далі відбувається заповнення решти значень змінної C[n] нулями за допомогою арифметичного циклу з параметром i (i приймає значення від k до n).
5. Наступним кроком відбувається три виклики функції output() для параметрів A,n ; B,n та C,n, за допомогою яких на екран виводяться три отримані послідовності A[n], B[n] та C[n].
6. Останнім кроком є пошук першого входження в третій масив C[n] елементу з кодом 52 за формулою number:=find\_52(C), яка викликає функцію find\_52().

* output(arr[], m) – функція, яка виводить переданий через параметр масив на екран, використовуючи арифметичний цикл з параметром i (i приймає значення від 1 до m включно) і виводячи на кожній ітерації змінну arr[], яка відповідає індексу i (arr[i]).
* find\_52(arr[]) – функція, яка проводить пошук першого входження в масив, переданий через параметр arr[] , елементу (символу) з кодом 52. Для цього функція виконує наступні дії:

1. Ініціалізує змінні u:=1 та r:=1 (r – булева змінна, тобто r присвоюється значення «істина»).
2. За допомогою ітераційного циклу з передумовою (умова: r==1) виконується лінійна перевірка кожного елементу масиву arr[], поки не буде знайдено перший елемент з кодом 52. У випадку успішного пошуку змінній num присвоюється індекс змінної arr[u] (u в кінці кожної ітерації збільшується на 1), тобто num:=u, а змінній r присвоюється значення 0 і відбувається вихід з ітераційного циклу.
3. Функція як результат повертає значення змінної num.

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

*Крок 1.* Визначимо основні дії.

*Крок 2.* Деталізуємо ініціалізацію змінних n та k.

*Крок 3.* Деталізуємо дію створення послідовностей A[n] та B[n].

*Крок 4.* Деталізуємо дію обчислення і-их членів послідовностей A і B.

*Крок 5.* Деталізуємо дію створення послідовності C[n].

*Крок 6.* Деталізуємо виведення послідовностей A[n], B[n] та C[n].

*Крок 7.* Деталізуємо пошук першого входження елементу з кодом 52.

*Крок 8.* Деталізуємо функцію output( ).

*Крок 9.* Деталізуємо функцію find\_52( ).

1. *Псевдокод алгоритму.*

*Крок 1*

**початок**

ініціалізація змінних n та k

створення послідовностей A[n] і B[n]

створення послідовності C[n]

виведення послідовностей A[n], B[n], C[n]

пошук першого входження елементу з кодом 52

виведення number

**кінець**

*Крок 2*

**початок**

n := 10

k := 1

створення послідовностей A[n] і B[n]

створення послідовності C[n]

виведення послідовностей A[n], B[n], C[n]

пошук першого входження елементу з кодом 52

виведення number

**кінець**

*Крок 3*

**початок**

n := 10

k := 1

**для** i від 1 до n

**повторити**

обчислення i-их членів послідовностей A і B

**все повторити**

створення послідовності C[n]

виведення послідовностей A[n], B[n], C[n]

пошук першого входження елементу з кодом 52

виведення number

**кінець**

*Крок 4*

**початок**

n := 10

k := 1

**для** iвід 1 до n

**повторити**

A[i] := 60 – 2 \* i

B[i] := 40 + 3 \* i

**все повторити**

створення послідовності C[n]

виведення послідовностей A[n], B[n], C[n]

пошук першого входження елементу з кодом 52

виведення number

**кінець**

*Крок 5*

**початок**

n := 10

k := 1

**для** i від 1 до n

**повторити**

A[i] := 60 – 2 \* i

B[i] := 40 + 3 \* i

**все повторити**

**для** i від 1 до n

**повторити**

**для** j від 1 до n

**повторити**

**якщо** (A[i] == B[j])

**то**

C[k] := A[i]

k := k + 1

**все якщо**

**все повторити**

**все повторити**

**для** i від k до n

**повторити**

C[i] := 0

**все повторити**

виведення послідовностей A[n], B[n], C[n]

пошук першого входження елементу з кодом 52

виведення number

**кінець**

*Крок 6*

**початок**

n := 10

k := 1

**для** i від 1 до n

**повторити**

A[i] := 60 – 2 \* i

B[i] := 40 + 3 \* i

**все повторити**

**для** i від 1 до n

**повторити**

**для** j від 1 до n

**повторити**

**якщо** (A[i] == B[j])

**то**

C[k] := A[i]

k := k + 1

**все якщо**

**все повторити**

**все повторити**

**для** i від k до n

**повторити**

C[i] := 0

**все повторити**

output(A, n)

output(B, n)

output(C, n)

пошук першого входження елементу з кодом 52

виведення number

**кінець**

*Крок 7*

**початок**

n := 10

k := 1

**для** i від 1 до n

**повторити**

A[i] := 60 – 2 \* i

B[i] := 40 + 3 \* i

**все повторити**

**для** i від 1 до n

**повторити**

**для** j від 1 до n

**повторити**

**якщо** (A[i] == B[j])

**то**

C[k] := A[i]

k := k + 1

**все якщо**

**все повторити**

**все повторити**

**для** i від k до n

**повторити**

C[i] := 0

**все повторити**

output(A, n)

output(B, n)

output(C, n)

number := find\_52(C)

виведення number

**кінець**

* 1. *Псевдокод допоміжних алгоритмів (функцій).*

*Крок 8*

**початок output(arr[], m)**

**для** i від 1 до m

**повторити**

виведення arr[i]

**все повторити**

**кінець output()**

*Крок 9*

**початок find\_52(arr[])**

u := 1

r :=1

**поки** r ==1

**повторити**

**якщо** (arr[u] == 52)

**то**

num := u

r :=0

**все якщо**

u++

**все повторити**

**повернути** num

**кінець find\_52()**

1. *Блок-схема алгоритму.*

*Крок 1 Крок 2*

*Крок 3 Крок 4*

*Крок 5 Крок 6*

*Крок 7*

* 1. *Блок-схеми допоміжних алгоритмів (функцій).*

 *Крок 8 Крок 9*

1. *Код програми (на мові програмування C++).*

#include <iostream>

#include <iomanip>

using namespace std;

void output(char[], int);

int find\_52(char[]);

int main()

{

const int n = 10;

int k = 0;

char A[n], B[n], C[n];

int number;

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

A[i] = 60 - 2 \* (i + 1);

B[i] = 40 + 3 \* (i + 1);

}

for (int i = 0; i < n; i++)

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

if (A[i] == B[j]) {

C[k] = A[i];

k++;

}

}

for (int i = k; i < n; i++) {

C[i] = 0;

}

cout << setw(20) << "The first massive: "; output(A, n);

cout << setw(20) << "The second massive: "; output(B, n);

cout << setw(20) << "The third massive: "; output(C, n);

number = find\_52(C);

cout << "The first occurrence of a number 52 has an index i = " << number << " (the " << number + 1 << " element of the third massive)" << endl;

system("pause");

return 0;

}

void output(char arr[], int m)

{

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

cout << setw(4) << arr[i];

}

cout << endl;

}

int find\_52(char arr[])

{

int num, u = 0;

bool r = 1;

while (r) {

if (arr[u] == 52) {

num = u;

r = 0;

}

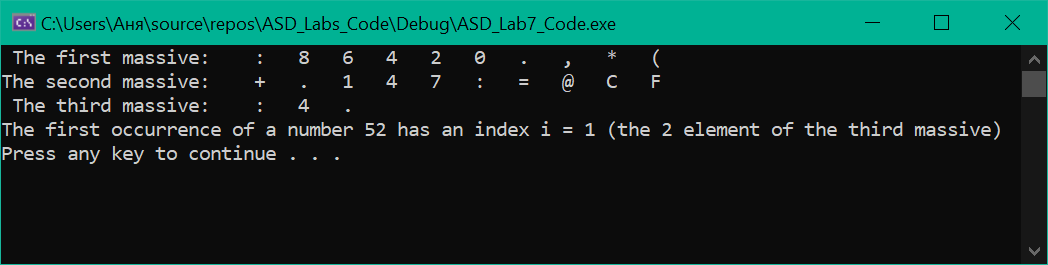
u++;

}

return num;

}

1. *Тестування програми.*

**

1. *Висновки.* На цій лабораторній роботі було досліджено методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та було набуто практичних навичок їх використання під час складання програмних специфікацій.

Побудований алгоритм було покладено на мову програмування C++ і написано код, який виконує поставлену задачу: знаходить дві послідовності з десяти символьних значень за даними формулами, ініціалізує третю послідовність рівними значеннями перших двох послідовностей та виконує операцію лінійного пошуку у третій послідовності і таким чином знаходить перше входження елементу з кодом 52, який за таблицею символів ASCII відповідає символу “4”. У результаті виконання коду отримали, що шуканий елемент має індекс 2. Дійсно, другий елемент третьої послідовності – “4”. Отже, алгоритм працює правильно.