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 арифметичних циклічних алгоритмів»

Варіант 27

Виконала: студентка ІП-21 Скрипець Ольга Олександрівна

Перевірила: Вєчерковська Анастасія Сергіївна

Київ 2022

**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.

**Завдання**

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 (табл. 1).

3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.

4. Обробки третьої змінної згідно з варіантом.





**Постановка задачі**

Результатом відпрацювання програми має бути сума кодів символів третього масиву, які більше 101. Вхідних даних не потрібно. Процедура Arr1(char \*array1) ініціює перший символьний масив символами, коди яких отримуються за правилом 95+i. Потім друга процедура Arr2(char \*array2) ініціює другий символьний масив, коди символів якого отримуються за правилом 105-i. Ініціювання третього масиву відбувається в функції fillArr3(\*array1, \*array2,\*array3), яка повертає його довжину. Для того, щоб ініціювати третій символьний масив рівними значеннями елементів з двох попередніх масивів, використаємо лінійний пошук, реалізуємо за допомогою складного циклу: зовнішній арифметичний та внутрішній арифметичний. Для отримання суми кодів символів третього масиву, які більше 101, використаємо арифметичний цикл.

**Побудова математичної моделі**

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Лічильник | Ціле | i | Лічильник |
| Масив з 10 елементів | символьний | array1[10] | Проміжне значення |
| Масив з 10 елементів | символьний | array2[10] | Проміжне значення |
| Масив з 10 елементів | символьний | array3[10] | Проміжне значення |
| Кількість елементів у 3 масиві | ціле | numberArr3 | Проміжне значення |
| Сума кодів символів 3 масиву, які більші за 101 | ціле | sum | Результат |
| Покажчик на перший символ масиву 1 | Покажчик символьного типу | \*array1 | Формальний параметр функції |
| Покажчик на перший символ масиву 2 | Покажчик символьного типу | \*array2 | Формальний параметр функції |
| Покажчик на перший символ третього масиву | Покажчик символьного типу | \*array3 | Формальний параметр функції |
| Кількість елементів третього масиву | Цілий | num | Результат виконання функції |
| Лічильник | Цілий | j | Параметр внутрішьного циклу |
| Кількість елементів переданого масиву | Цілий | number | Формальний параметр функції |

**Результат**

Крок 1. Визначимо основні дії

Крок 2. Деталізуємо процедуру Arr1

Крок 3. Деталізуємо процедуру Arr2

Крок 4. Деталізуємо підпрограму printArr

Крок 5. Деталізуємо функцію fillArr3

**Псевдокод алгоритму**

**main:**

**початок**

array1[10]={}

array2[10]={}

array3[10]={}

sum=0

**виклик процедури** Arr1(array1)

**виклик процедури** Arr2(array2)

numberArr3=fillArr3(array1, array2, array3)

**повторити**

**для** i **від** 0 **до** numberArr3 **з кроком** 1

**якщо** int(array3[i])>101

**то**

sum=sum+array[i]

**все якщо**

**все повторити**

**виклик процедури** printArray(array1, 10)

**виклик процедури** printArray(array2, 10)

**виклик процедури** printArray(array3, numberArr3)

**виведення:** sum

**кінець**

**Arr1**(\*array1):

**повторити**

**для** i **від** 0 **до** 9 **з кроком** 1

array1[i]=95+i

**все повторити**

**кінець**

**Arr2**(\*array2):

**повторити**

**для** i **від** 0 **до** 9 **з кроком** 1

array2[i]=105+i

**все повторити**

**кінець**

**printArray**(\*array1, number):

**повторити**

**для** i **від** 0 **до** number **з кроком** 1

**виведення:** array1[i]

**все повторити**

**кінець**

**fillArr3**(\*array1, \*array2, \*array3):

num=0

**повторити**

**для** i **від** 0 **до** 9 **з кроком** 1

**повторити**

**для** j **від** 0 **до** 9 **з кроком** 1

**якщо** array1[i]==array2[j]

**то**

array3[num]=array1[i]

num=num+1

**все якшо**

**все повторити**

**все повторити**

**return** num

**Код алгоритму**

#include <stdio.h>  
  
void Arr1(char \*array1);  
void Arr2(char \*array2);  
void printArr(char \*array1, int number);  
int fillArr3(char \*array1, char \*array2, char \*array3);  
  
int main(){  
 char array1[10]={}, array2[10]={}, array3[10]={};  
 int numberArr3, sum=0;  
  
 Arr1(array1);  
 Arr2(array2);  
 numberArr3=fillArr3(array1, array2, array3);  
  
 for(int i=0; i<=numberArr3; i++){  
 if(array3[i]>101){  
 sum+=array3[i];  
 }  
 }  
  
 printf("Array1: ");  
 printArr(array1, 10);  
 printf("\nArray2: ");  
 printArr(array2, 10);  
 printf("\nArray3: ");  
 printArr(array3, numberArr3);  
 printf("\nResult: %d", sum);  
  
 return 0;  
}  
  
void Arr1(char \*array1){  
 for(int i=0; i<10; i++){  
 array1[i]=95 + i;  
 }  
}  
  
void Arr2(char \*array2){  
 for(int i=0; i<10; i++){  
 array2[i]=105 - i;  
 }  
}  
  
int fillArr3(char \*array1, char \*array2, char \*array3){  
 int num=0;  
 for(int i=0; i<10; i++){  
 for(int j=0; j<10; j++){  
 if(array1[i]==array2[j]){  
 array3[num]=array1[i];  
 num++;  
 }  
 }  
 }  
 return num;  
}  
  
void printArr(char \*array1, int number){  
 for(int i=0; i<number; i++){  
 printf("%d ", array1[i]);  
 }  
}

**Випробування алгоритму**

|  |  |
| --- | --- |
| Блок | Дія |
|  | **початок** |
| 1 | array1[10]={} |
| 2 | array2[10]={} |
| 3 | array3[10]={} |
| 4 | sum=0 |
| 5 | **виклик процедури** Arr1(array1) |
| 6(1 ітерація циклу) | i=0; array1[0]=char(95) |
| 7(2 ітерація циклу) | i=1; array1[1]=char(96) |
| 8(3 ітерація циклу) | i=2; array1[2]=char(97) |
| 9(4 ітерація циклу) | i=3; array1[3]=char(98) |
| 10(5 ітерація циклу) | i=4; array1[4]=char(99) |
| 11(6 ітерація циклу) | i=5; array1[5]=char(100) |
| 12(7 ітерація циклу) | i=6; array1[6]=char(101) |
| 13(8 ітерація циклу) | i=7; array1[7]=char(102) |
| 14(9 ітерація циклу) | i=8; array1[8]=char(103) |
| 15(10 ітерація циклу) | i=9; array1[9]=char(104) |
| 16 | **виклик процедури** Arr2(array2) |
| 17(1 ітерація циклу) | i=0; array2[0]=char(105) |
| 18(2 ітерація циклу) | i=1; array2[1]=char(104) |
| 19(3 ітерація циклу) | i=2; array2[2]=char(103) |
| 20(4 ітерація циклу) | i=3; array2[3]=char(102) |
| 21(5 ітерація циклу) | i=4; array2[4]=char(101) |
| 22(6 ітерація циклу) | i=5; array2[5]=char(100) |
| 23(7 ітерація циклу) | i=6; array2[6]=char(99) |
| 24(8 ітерація циклу) | i=7; array2[7]=char(98) |
| 25(9 ітерація циклу) | i=8; array2[8]=char(97) |
| 26(10 ітерація циклу) | i=9; array2[9]=char(96) |
| 27 | **виклик функції** fillArr3(array1, array2, array3) |

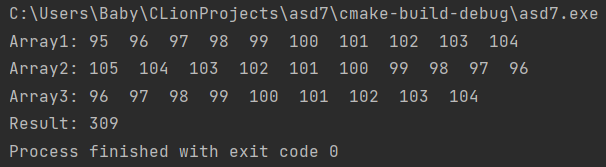
У результаті виклику функціїfillArr3(array1, array2, array3) третій масив Arr3 буде заповнено таким чином:

|  |  |  |
| --- | --- | --- |
| Array1(i від 0 до 9) | Array2(j від 0 до 9) | Array3 |
| **95** | **105** | **96** |
| **96** | **104** | **97** |
| **97** | **103** | **98** |
| **98** | **102** | **99** |
| **99** | **101** | **100** |
| **100** | **100** | **101** |
| **101** | **99** | **102** |
| **102** | **98** | **103** |
| **103** | **97** | **104** |
| **104** | **96** |  |

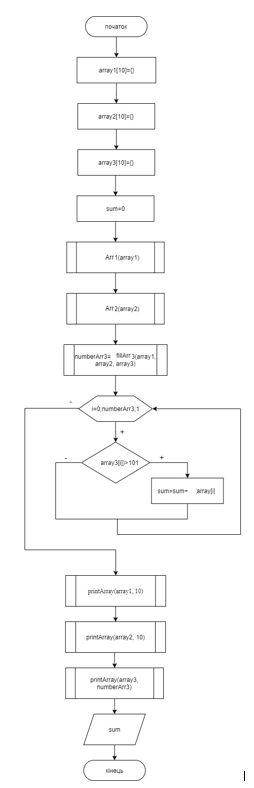
Продовження:

|  |  |
| --- | --- |
|  | numberArr3=9 |
| (1 ітерація циклу) | **i=0** |
| array3[i]>101 | (96>101)=false |
| (2 ітерація циклу) | **i=1** |
| array3[i]>101 | (97>101)=false |
| (3 ітерація циклу) | **i=2** |
| array3[i]>101 | (98>101)=false |
| (4 ітерація циклу) | **i=3** |
| array3[i]>101 | (99>101)=false |
| (5 ітерація циклу) | **i=4** |
| array3[i]>101 | (100>101)=false |
| (6 ітерація циклу) | **i=5** |
| array3[i]>101 | (101>101)=false |
| (7 ітерація циклу) | **i=6** |
| array3[i]>101 | (102>101)=true |
| sum=sum+array[i] | **sum=0+102=102** |
| (8 ітерація циклу) | i=7 |
| array3[i]>101 | **(103>101)=true** |
| sum=sum+array[i] | sum=102+103=205 |
| (9 ітерація циклу) | **i=8** |
| array3[i]>101 | (104>101)=true |
| sum=sum+array[i] | **sum=205+104=309** |
|  | виклик процедури  printArr(array1, 10) |
|  | **виклик процедури**  **printArr(array2, 10)** |
|  | виклик процедури  printArr(array3, numberArr3) |
|  | **виведення: 309** |
|  | **кінець** |

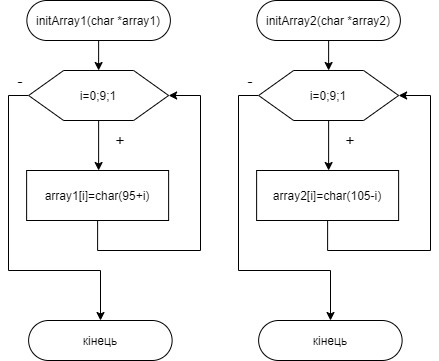
**Тестування програми**

****

**Блок схема алгоритму**

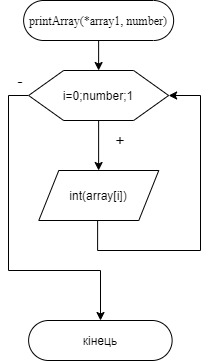
**Main:**

**Arr1(\*array1): Arr2(\*array2):**

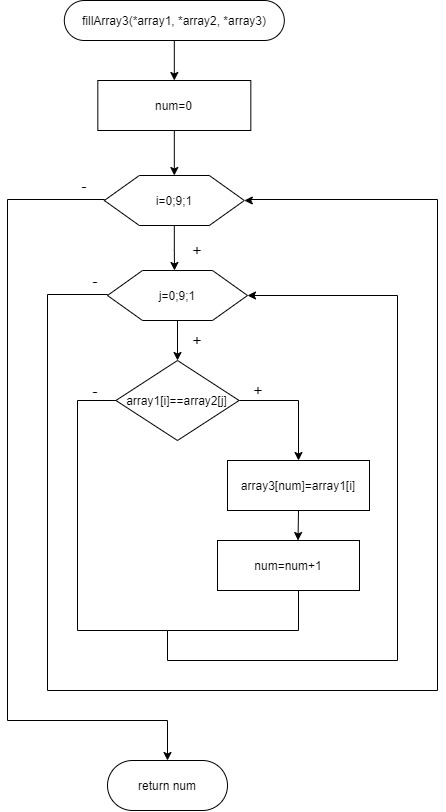


****

**printArr(\*array1, number):**



**fillArr3(\*array1, \*array2, \*array3):**



**Висновки**

Ми дослідили методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та набули практичних навичок їх використання під час складання програмних специфікацій. У результаті виконання лабораторної роботи ми отримали алгоритм який ініціює два символьні масиви числовими послідовностями за заданими правилами: код елемента першого масиву: 95+i; код елемента другого масиву: 105-i. За це відповідають процедури Arr1(\*array1) та Arr2(\*array2). Функція fillArr3(\*array1, \*array2, \*array3) ініціює третій масив однаковими значеннями двох перших та повертає їх кількість (довжину третього масиву). Для отримання суми кодів символів третього масиву, які більше 101, використала арифметичний цикл.