
Алгоритми та структури даних. Основи алгоритмізації

Додаток 1

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 лінійних алгоритмів»

Варіант 18

Виконав студент Король О.В.

(шифр, прізвище, ім'я, по батькові)

Перевірила Вечерковська А.С.

(прізвище, ім'я, по батькові)

Київ 2021

Алгоритми та структури даних. Основи алгоритмізації

Лабораторна робота 7

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.

Індивідуальне завдання

18	$60 - 2 * i$	$40 + 3 * i$	Перше входження елементу з кодом 52
----	--------------	--------------	-------------------------------------

Варіант 18 Постановка задачі

Результатом є виведення першого елемента третього масиву, який дорівнює 52

Побудова математичної моделі

Складемо таблицю імен змінних та функцій

Змінна	Тип	Ім'я	Призначення
Перший масив	Символьний	array1	Проміжні дані
Другий масив	Символьний	array2	Проміжні дані
Третій масив	Символьний	array3	Проміжні дані
Індекс масивів	цілі	i	Ініціювання першого масиву
Результат	Цілочисельний	num	Результат

arr1 - функція для ініціювання першого масиву, приймає як параметр перший масив. arr2 - функція для ініціювання другого масиву, приймає як параметр другий масив. arr3 - функція для ініціювання третього масиву, приймає як параметр перший, другий і третій масиви. array1, array2, array3 – 10-елементні масиви.

Res - функція, що обчислює змінну res, приймає як параметр третій масив.

Алгоритми та структури даних. Основи алгоритмізації

res - змінна, що містить кількість елементів третього масиву, коди яких діляться на 5.

Розв'язання

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми. Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо функцію arr1.

Крок 3. Деталізуємо функцію arr2.

Крок 4. Деталізуємо функцію arr3.

Крок 5. Деталізуємо функцію Res.

Псевдокод

Крок 1

Підпрограма

output

все підпрограма

початок

int i,j

const int n = 10;

char A[n]

char B[n]

char C[n]

виведення num

кінець

Крок 2

Підпрограма output

повторити для i від 1 до n,

збільшувати на 1

все підпрограма

початок

int i,j

const int n = 10;

char A[n];

char B[n];

char C[n]

int num

виведення num

кінець

Алгоритми та структури даних. Основи алгоритмізації

Крок 3

Підпрограма output

повторити для i від 1 до n,

збільшувати на 1

все підпрограма

початок

int i,j

const int n = 10;

char A[n];

char B[n];

char C[n]

повторити для i від 1 до n,

збільшувати на 1

A[i] = 60 - 2 * i

B[i] = 40 + 3 * i

Все повиорити

повторити для i від 0 до n,

збільшувати на 1

повторити для j від 0 до n,

збільшувати на 1

ЯКЩО A[i] == B[j]

то

C[k] = A[i]

K + 1

все повторити

Bool l = 1

Поки bool l = 1

Якщо

C[i] == 52

num = i

l = 0;

все повторити

виведення num

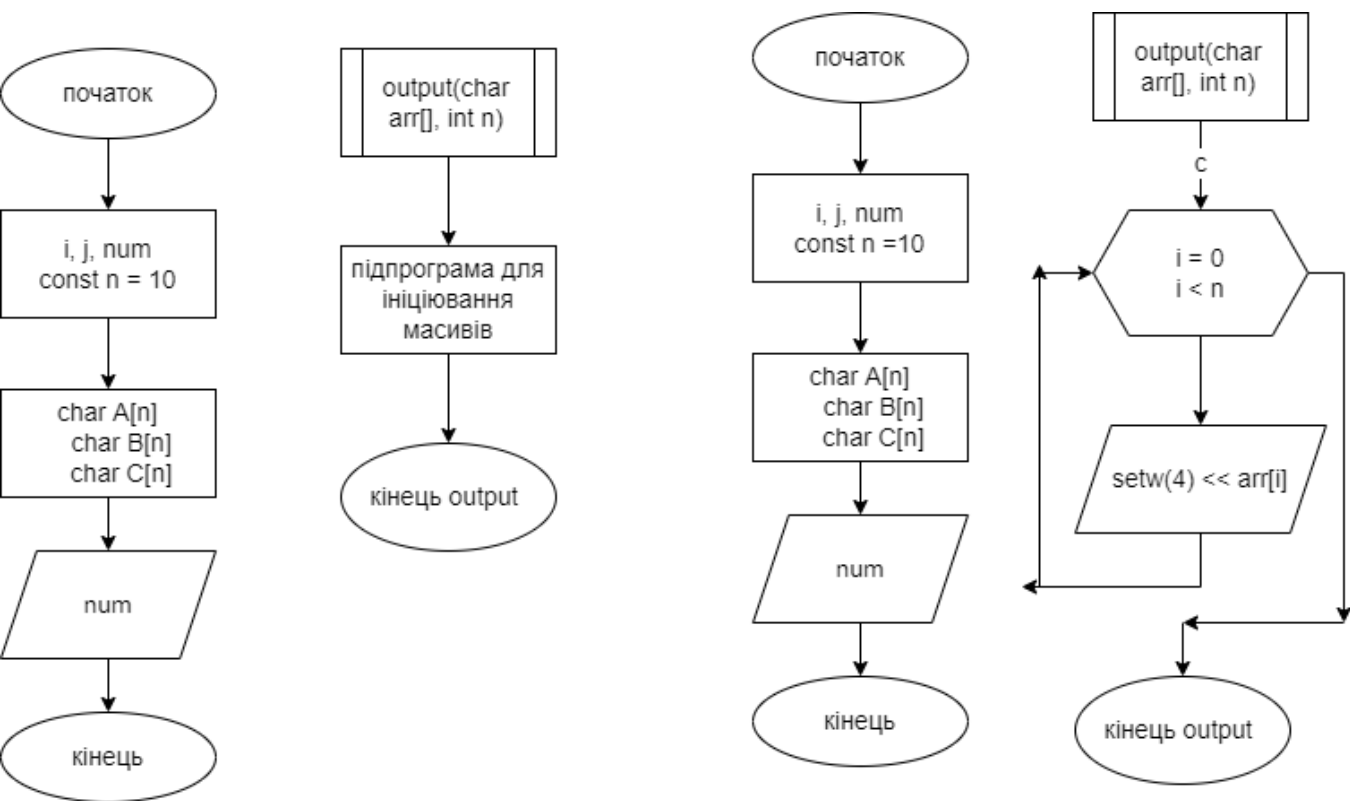
кінець

Блок-схема

Крок 1

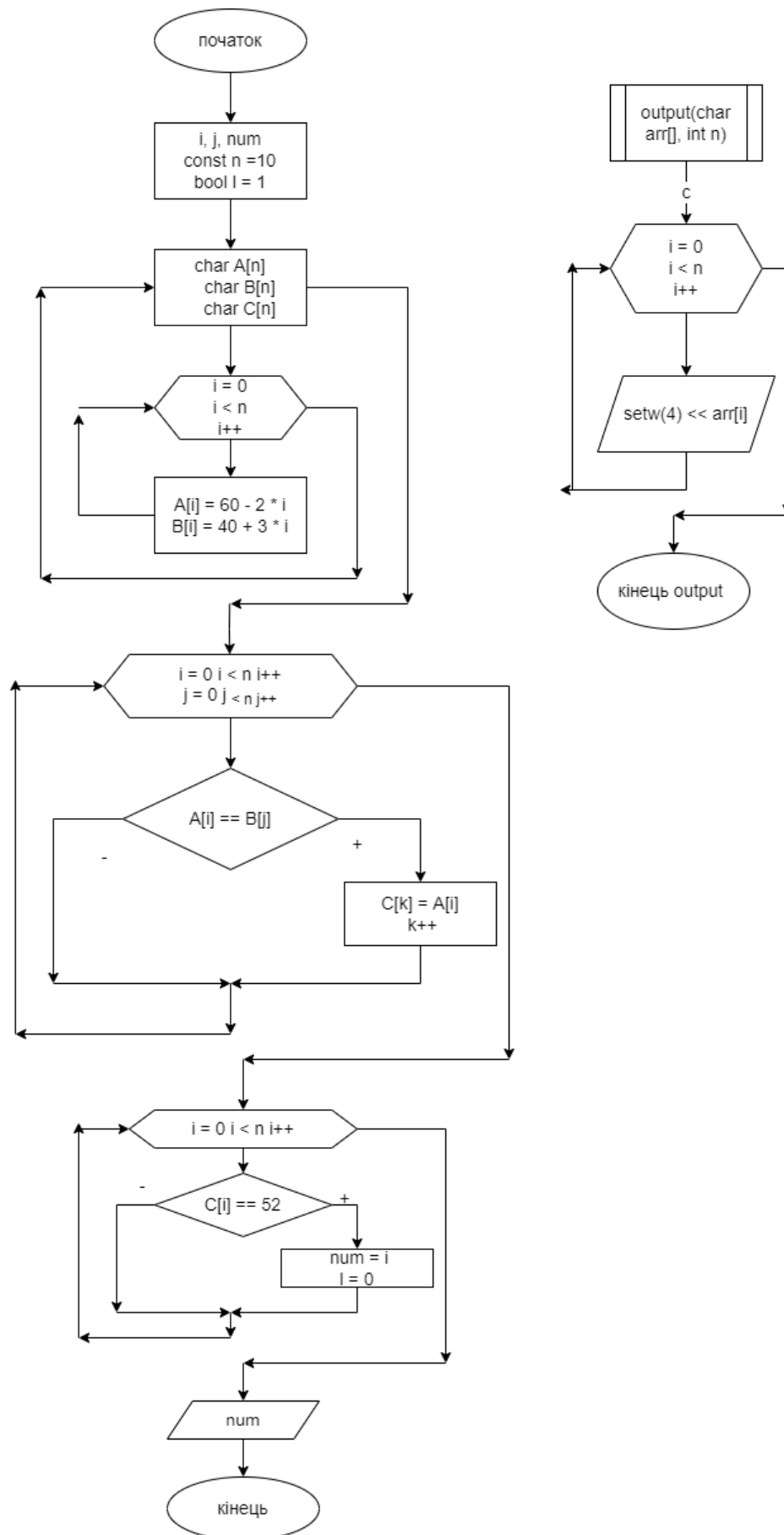
Крок 2

Алгоритми та структури даних. Основи алгоритмізації



Крок 3

Алгоритми та структури даних. Основи алгоритмізації



Алгоритми та структури даних. Основи алгоритмізації

Результат коду

```
#include <iomanip>
#include <iostream>

using namespace std;

void output(char arr[], int n)
{
    for (int i = 0; i < n; i++)
        cout << setw(4) << arr[i];

    cout << endl;
}

int main()
{
    int i, j;
    const int n = 10;
    char A[n];
    char B[n];
    char C[n];
    for (i = 0; i < n; i++)
    {
        A[i] = 60 - 2 * i;
        B[i] = 40 + 3 * i;
    }
    int k = 0;
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
        {
            if (A[i] == B[j])
            {
                C[k] = A[i];
                k++;
            }
        }
    for (i = k; i < n; i++)
```

Алгоритми та структури даних. Основи алгоритмізації

```
{
    A[i] = 60 - 2 * i;
    B[i] = 40 + 3 * i;
}

int k = 0;
for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
    {
        if (A[i] == B[j])
        {
            C[k] = A[i];
            k++;
        }
    }

for (i = k; i < n; i++)
    C[i] = 0;
cout << setw(20) << "First massive: "; output(A, n);
cout << setw(20) << "Second massive: "; output(B, n);
cout << setw(20) << "Third massive: "; output(C, n);
int num;
bool l = 1;
i = 0;
while (l)
{
    if (C[i] == 52)
    {
        num = i;
        l = 0;
    }
    i++;
}
cout << "The first coincidence of a number 52 has an index i = " << num << " (the " << num + 1 << " element of the third massive)" << endl;
system("pause");
}
```



```
C:\Users\valik\source\repos\acdlab7\Debug\acdlab7.exe

First massive:  < : 8 6 4 2 0 . , *
Second massive: ( + . 1 4 7 : = @ C
Third massive:  : 4 .

The first coincidence of a number 52 has an index i = 1 (the 2 element of the third massive)
Press any key to continue . . .
```

Висновки

Алгоритми та структури даних. Основи алгоритмізації

Я дослідив методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набув практичних навичок їх використання під час складання програмних специфікацій. Побудував мат. модель, псевдокод та блок схему. Протестував алгоритм.