

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 № 1 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«Дослідження алгоритмів розгалуження»

Варіант 17

Виконав студент ПІ-12, Коновалюк Іванна Леонідівна

Перевірив

(прізвище, ім'я, по батькові)

Київ 2021

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.

Варіант 17

	Вираз для обчислення елемента		Знайти
	1-го масиву	2-го масиву	
17	$5 * i + 25$	$55 - 5 * i$	Добуток елементів, коди яких менше 82

Постановка задачі. Заданий алгоритм повинен створити три змінні індексованого типу, ініціювати перші 2 змінні за формулами $5i+25$ та $55-5i$, ініціювати третю змінну спільними значеннями двох попередніх та знайти добуток елементів з третьої змінної, коди яких менше 82.

Математична модель.

Змінна	Тип	Ім'я	Призначення
Розмірність масивів	Натуральний	z	Початкові дані
Перший масив	Символьний	F	Проміжні дані
Другий масив	Символьний	S	Проміжні дані
Третій масив	Символьний	T	Проміжні дані
Добуток кодів	Натуральний	product	Проміжні дані
Результат добутка	Натуральний	result	Кінцеві дані

Власні функції

input1(ціле число) – повертає перший символьний масив.

input2(ціле число) – повертає другий символьний масив.

input3(ціле число, одновимірний символьний масив, одновимірний символьний масив) – повертає третій символьний масив згенерований з двох перших масивів.

product(одновимірний символьний масив) – повертає добуток елементів, коди яких менше 82.

output(одновимірний символьний масив) – виведення масивів.

Розв'язання.

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію визначення першого масиву з використанням функції.

Крок 3. Деталізуємо дію визначення другого масиву з використанням функції.

Крок 4. Деталізуємо дію визначення третього масиву з використанням функції.

Крок 5. Деталізуємо дію виведення трьох масивів з використанням функції.

Крок 6. Деталізуємо дію обрахунку добутку елементів третього масиву з використанням функції.

Псевдокод алгоритму.

Крок 1

Початок

z = 10

Визначення першого масиву.

Визначення другого масиву.

Визначення третього масиву.

Виведення трьох масивів.

Обрахунок добутків елементів третього масиву.

Виведення result
Кінець

Крок 2.

Початок

z = 10

input1(F)

Визначення другого масиву.

Визначення третього масиву.

Виведення трьох масивів.

Обрахунок добутків елементів третього масиву.

Виведення result

Кінець

Крок 3.

Початок

z = 10

input1(F)

input2(S)

Визначення третього масиву.

Виведення трьох масивів.

Обрахунок добутків елементів третього масиву.

Виведення result

Кінець

Крок 3.

Початок

z = 10

input1(F)

input2(S)

input3(T, F, S)

Виведення трьох масивів.

Обрахунок добутків елементів третього масиву.

Виведення result

Кінець

Крок 4.

Початок

z = 10

input1(F)

input2(S)

input3(T, F, S)

output(F)

output(S)

output(T)

Обрахунок добутків елементів третього масиву.

Виведення result

Кінець

Крок 5.

Початок

z = 10

```
input1(F)
input2(S)
input3(T, F, S)
output(F)
output(S)
output(T)
result = product(T)
Виведення result
```

Кінець

Підпрограма input1(array)

```
повторити
для i від 1 до z
    array[i] := 25 + 5 * (i + 1)
все повторити
```

Все підпрограма

Підпрограма input2(array)

```
повторити
для i від 1 до z
    array[i] := 55 - 5 * (i + 1)
все повторити
```

Все підпрограма

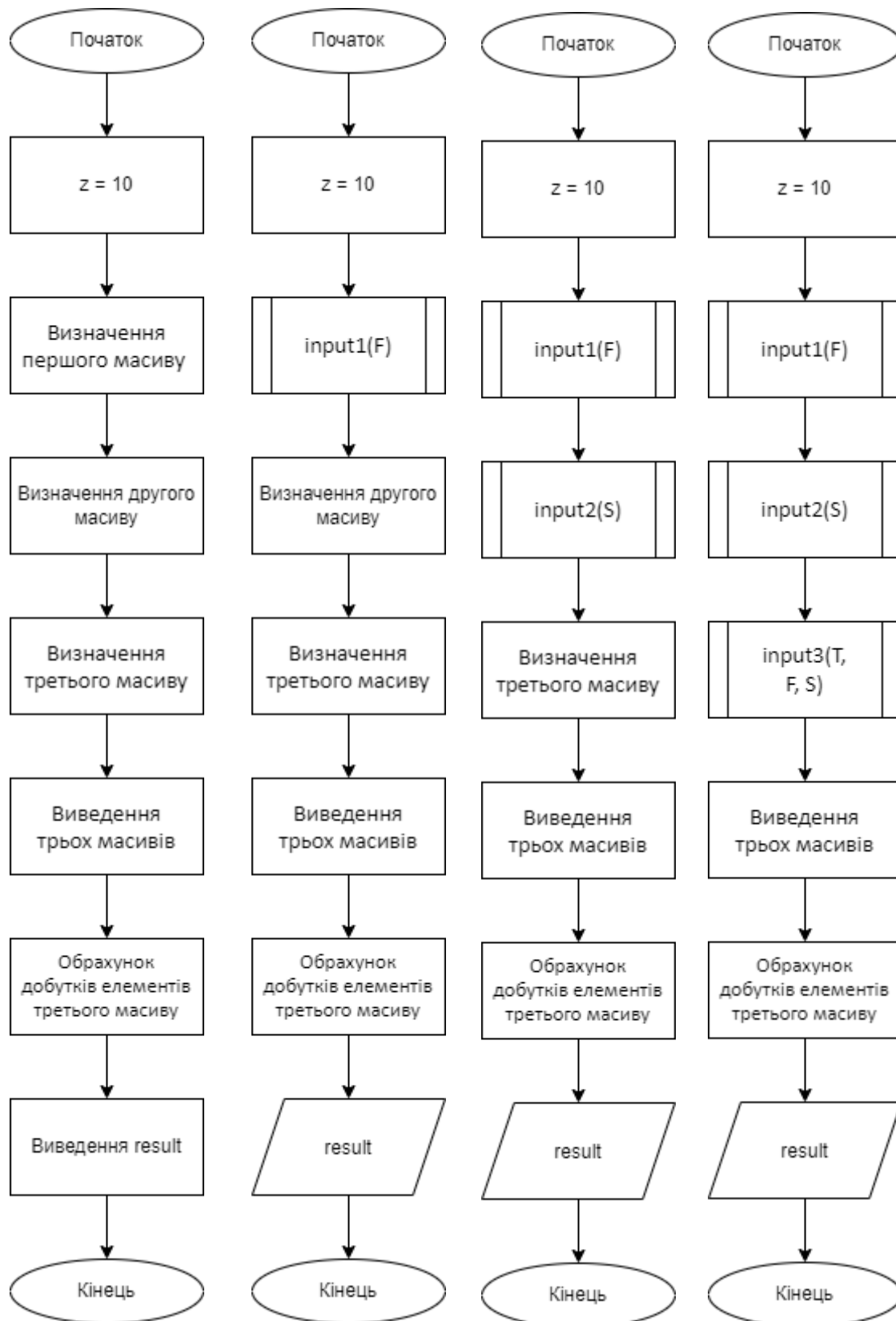
Підпрограма input3(array3, array1, array2)

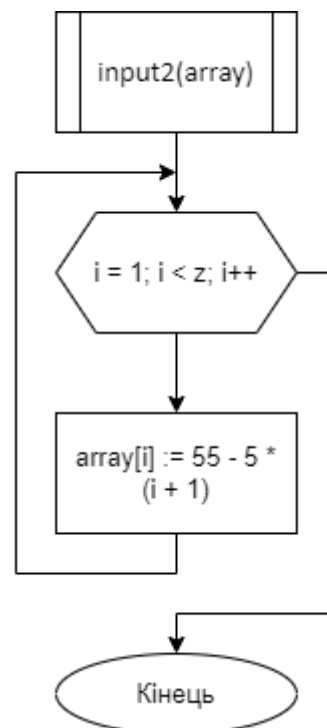
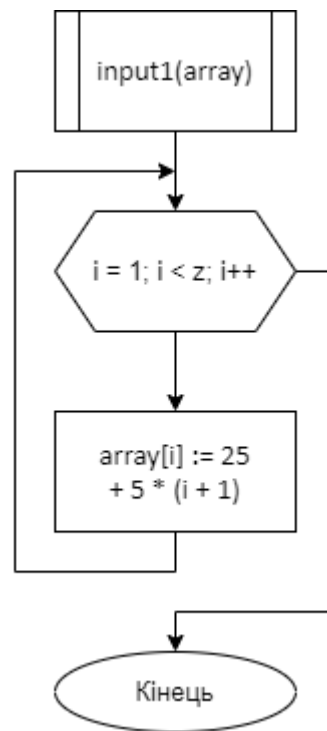
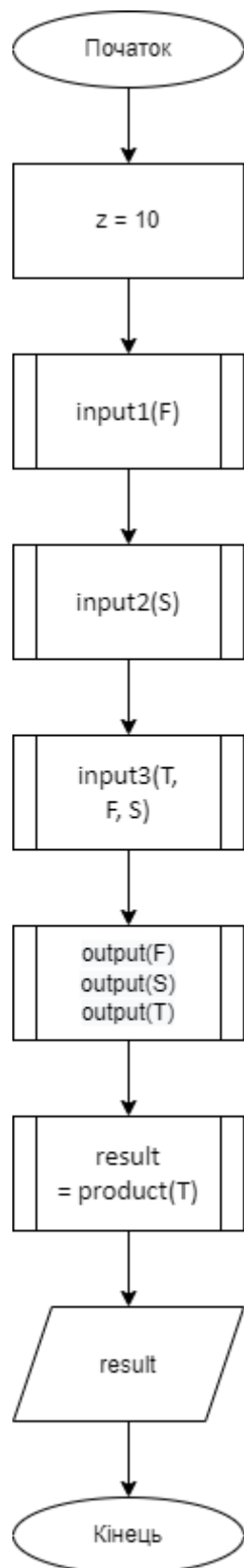
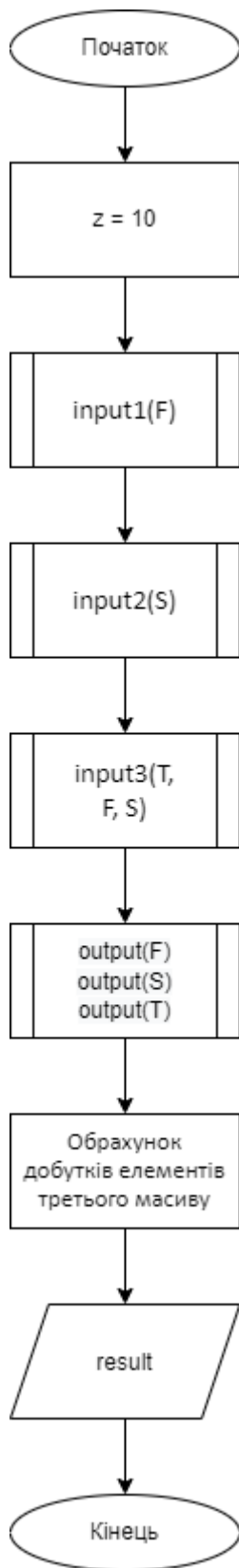
```
повторити
для i від 1 до z
    повторити
    для j від 1 до z
        якщо array1[i] == array2[j]
            то
                array3[i] := array2[j]
                T[i] := array3[i]
        все якщо
    все повторити
все повторити
```

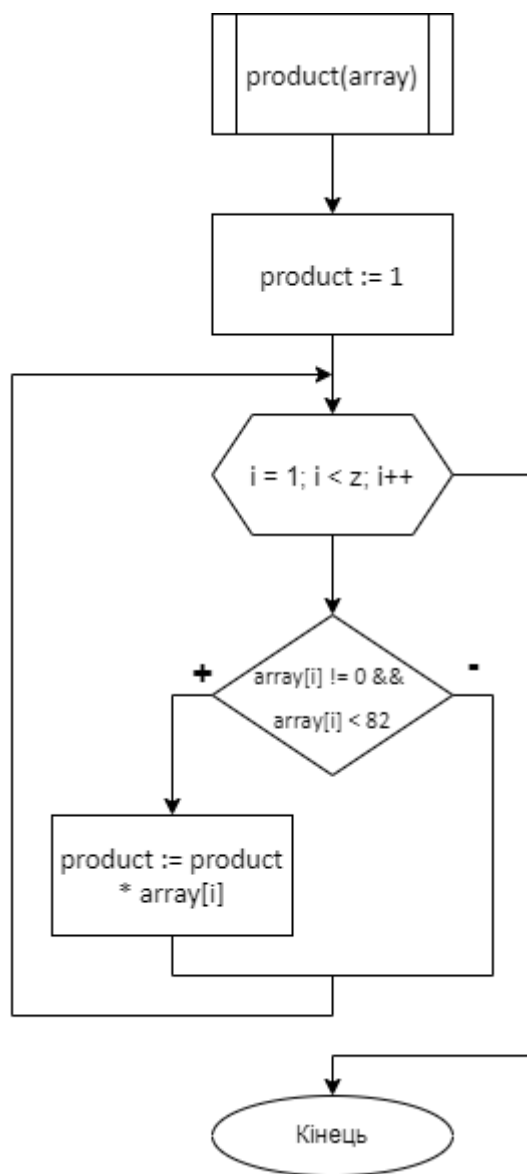
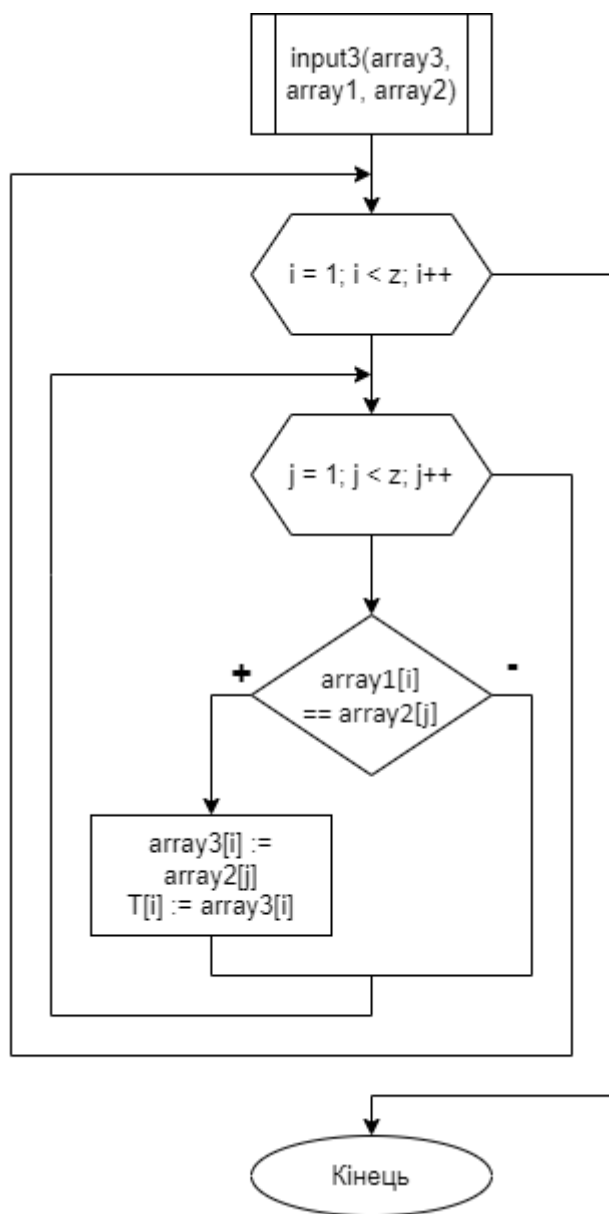
Підпрограма product(array)

```
product := 1
повторити
для i від 1 до z
    якщо array[i] != 0 && array[i] < 82
        то
            product := product * array[i]
    все якщо
все повторити
```

Блок-схема алгоритму.







Код програми

```
#include <iostream>
using namespace std;

const int z = 10;
char F[z], S[z], T[z];

void input1(char[]);
void input2(char[]);
void input3(char[], char[], char[]);
char product(char[]);
void output(char[]);

int main()
{
    int result;
    input1(F);
    input2(S);
    input3(T, F, S);
    cout << "First array: \n";
    output(F);
    cout << "Second array: \n";
    output(S);
    cout << "Third array: \n";
    output(T);
    result = product(T);
    cout << "Product: \n" << result;
}

void input1(char array[])
{
    for (int i = 0; i < z; i++)
    {
        array[i] = 25 + 5 * (i + 1);
    }
}

void input2(char array[])
{
    for (int i = 0; i < z; i++)
    {
        array[i] = 55 - 5 * (i + 1);
    }
}

void input3(char array3[], char array1[], char array2[])
{
    for (int i = 0; i < z; i++)
    {
        for (int j = 0; j < z; j++)
        {
            if (array1[i] == array2[j])
            {
                array3[i] = array2[j];
                T[i] = array3[i];
            }
        }
    }
}
```



```

char product(char array[])
{
    int product = 1;
    for (int i = 0; i < z; i++)
    {
        if (array[i] != 0 && array[i] < 82)
        {
            product = product * array[i];
        }
    }
    return product;
}

void output(char array[])
{
    for (int i = 0; i < z; i++)
    {
        cout << array[i] << " ";
    }
    cout << "\n";
}

```

Випробування коду

```

Консоль отладки Microsoft Visual Studio

First array:
▲ # < - 2 ? < n F K
Second array:
2 - < # ▲ ↓ ¶ *
Third array:
▲ # < - 2
Product:
-96
C:\Users\HP-HP\source\repos\Labs<>\Debug\Черновик.exe (процесс 25032) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно:

```

Висновок. У результаті лабораторної роботи було досліджено методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набуто практичні навички їх використання під час складання програмних специфікацій. Було поставлено задачу, побудовано математичну модель, розроблено алгоритм її вирішення у вигляді псевдокоду, який було переведено на блок-схему. Алгоритм успішно генерує два одновимірні символьні масиви за формулами, генерує третій масив з сільних елементів попередніх масивів та знаходить добуток елементів, коди яких менше 82.