

Основи програмування – 1. Алгоритми та структури

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

« Дослідження алгоритмів пошуку та сортування»»
Варіант 17

Виконав студент ІП-15 Куркчі Юрій Сергійович
(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота 8 Дослідження алгоритмів пошуку та сортування»

Мета – дослідити алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Індивідуальне завдання

Варіант 17

1.Опису змінної індексованого типу (двовимірний масив) згідно з варіантом (табл. 1).

2.Ініціювання змінної, що описана в п.1 даного завдання.

3.Створення нової змінної індексованого типу (одновимірний масив) та її ініціювання значеннями, що обчислюються згідно з варіантом (табл. 1).

Табл. 1

17	6 x 5	Цілий	Із суми додатних значень елементів рядків двовимірного масиву. Відсортувати методом вставки за зростанням.
----	-------	-------	--

Побудова математичної моделі

Таблиця змінних

Змінна	Тип	Ім'я	Призначення
Кількість рядків	Натуральний	Z	Початкові дані
Кількість стовпців	Натуральний	G	Проміжні дані
Вхідна матриця	Цілий[Z][G]	M	Вхідні данні
Одновимірний масив	Цілий[Z]	mult	Результат
Ітератор	Цілий	i	Проміжні дані
Добуток кодів	Натуральний	mul	Проміжні дані

Ітератор	Цілий	item	Проміжні дані
Ітератор	Цілий	j	Проміжні дані
Тимчасова змінна для перестановки елементів	Цілий	temp	Проміжні дані

За умовою створимо матрицю розміром 6 на 5 та заповнимо її числами (у програмі генеруються випадкові числа). Потім знаходимо суми додатних значень елементів рядків двовимірного масиву і записуємо їх у новий масив. Далі йде сортування вставкою.

Власні функції

- 1 input(двовимірний масив) – створює матрицю
- 2 output(двовимірний масив) –виводить матрицю
- 3 sort(двовимірний масив) – створює й виводить одновимірний масив .
- 4 sort2(двовимірний масив) – сортує одновимірний масив за зростання.

Розв'язання

1. Визначимо основні дії.
2. Деталізуємо дію визначення матриці.
3. Деталізуємо дію та виведення матриці.
4. Деталізуємо дію визначення та виведення одновимірного масиву
5. Деталізуємо дію сортування одновимірного масиву
6. Деталізуємо дію виведення відсортованого одновимірного масиву.

Псевдокод алгоритму

Крок 9

Початок

Z:=6

G:=5

Input(M)

output(M);

sort(M);

sort2(M);

повторити

для i від 1 до Z

вивід mult[i]

Кінець

Підпрограма

input(masiv[][G])

повторити

для i від 1 до Z

повторити

для j від 1 до G

masiv[i][j] = rand() %

17 - 8

все повторити

все повторити

Все підпрограма

Підпрограма

output(masiv[][G])

повторити

для i від 1 до Z

повторити

для j від 1 до G

вивід masiv[i][j]

все повторити

все повторити

Все підпрограма

Підпрограма

sort(masiv[][G])

повторити

для i від 1 до Z

mul=1

повторити

для j від 1 до

якщо masiv[i][j]>0

то mul=mul*masiv[i][j]

все якщо

все повторити

mult[i]=mul

виведення mult[i]

все повторити

Все підпрограма

Підпрограма

Sort2(masiv[])

повторити

для i від 1 до Z

temp = mult[i];

item = i-1;

поки item >= 0 && mult[item] > temp

mult[item + 1] = mult[item]

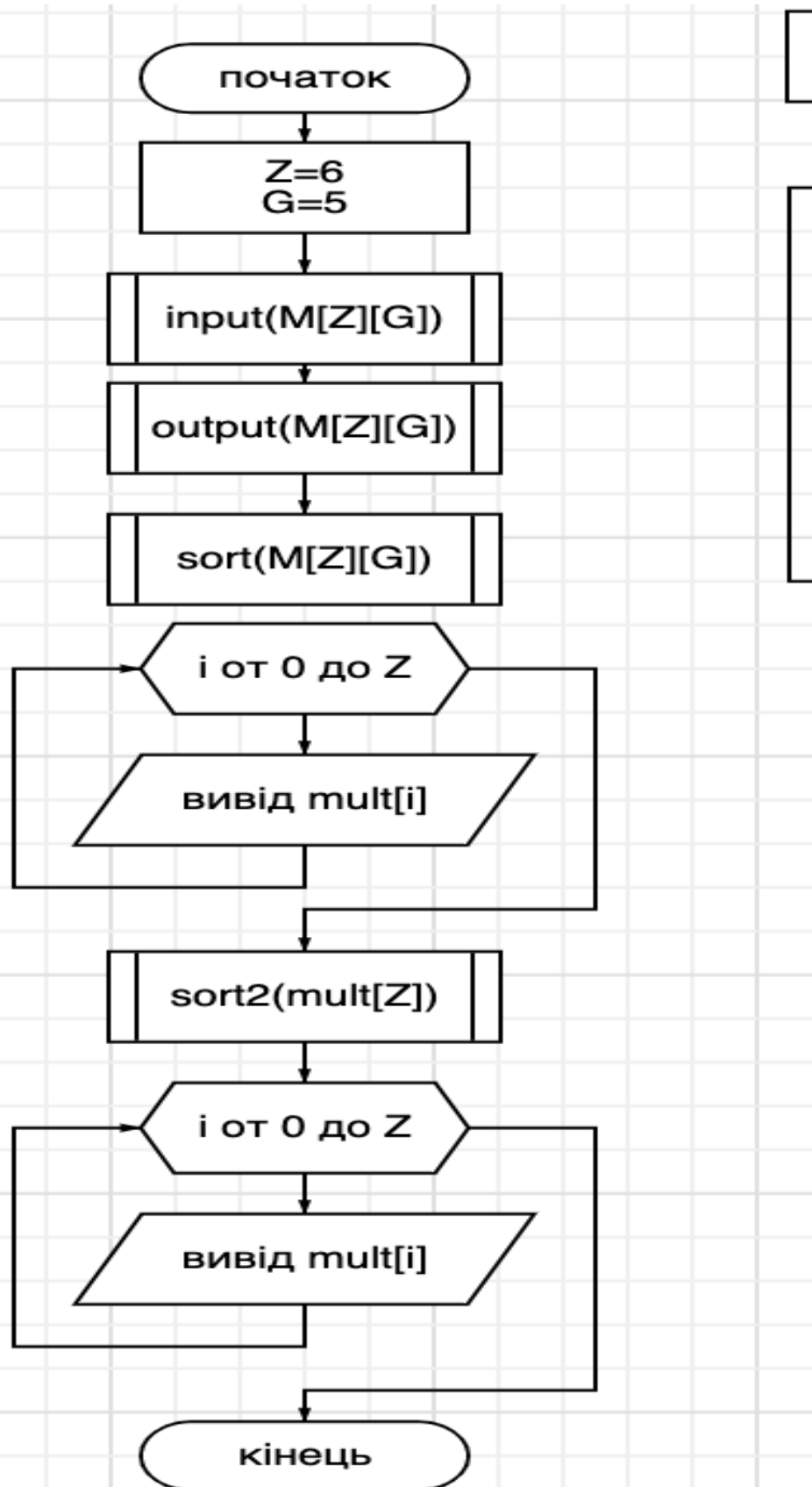
mult[item] = temp

item—

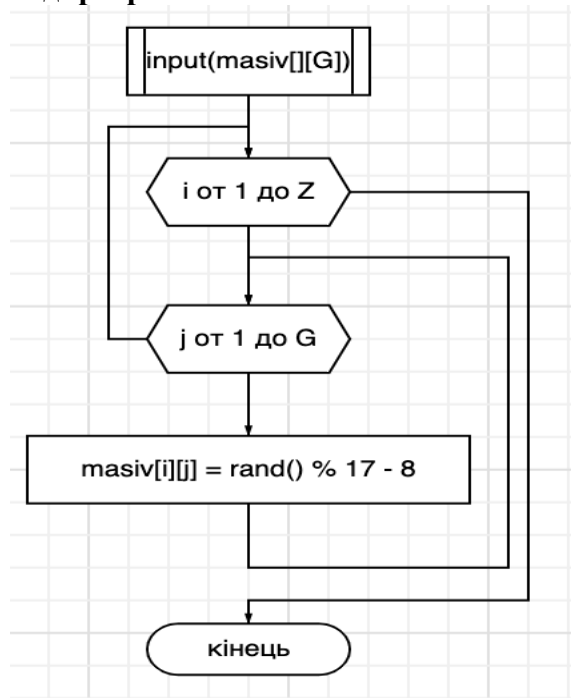
все повторити

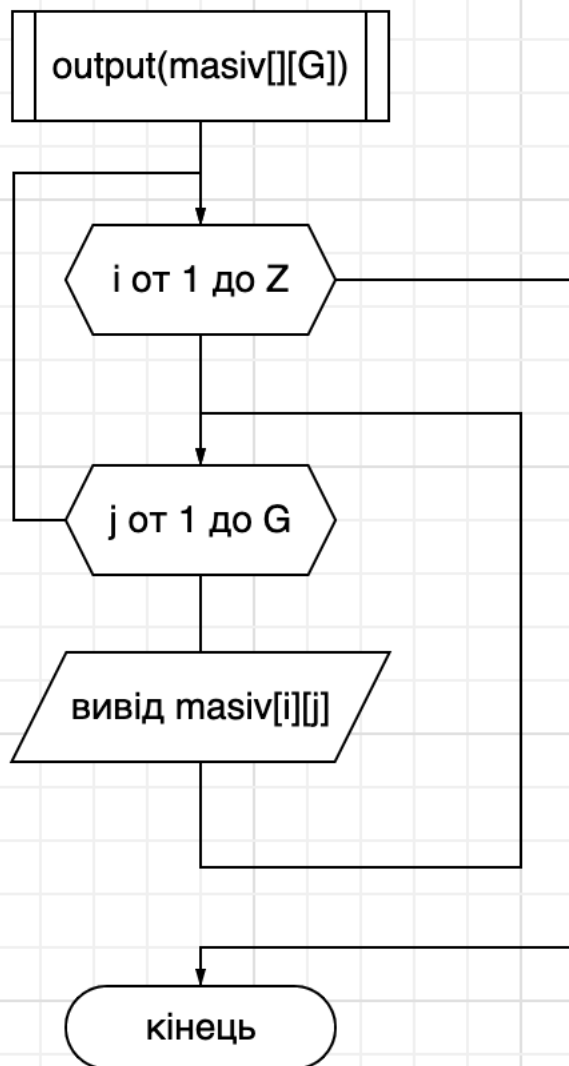
Все підпрограма

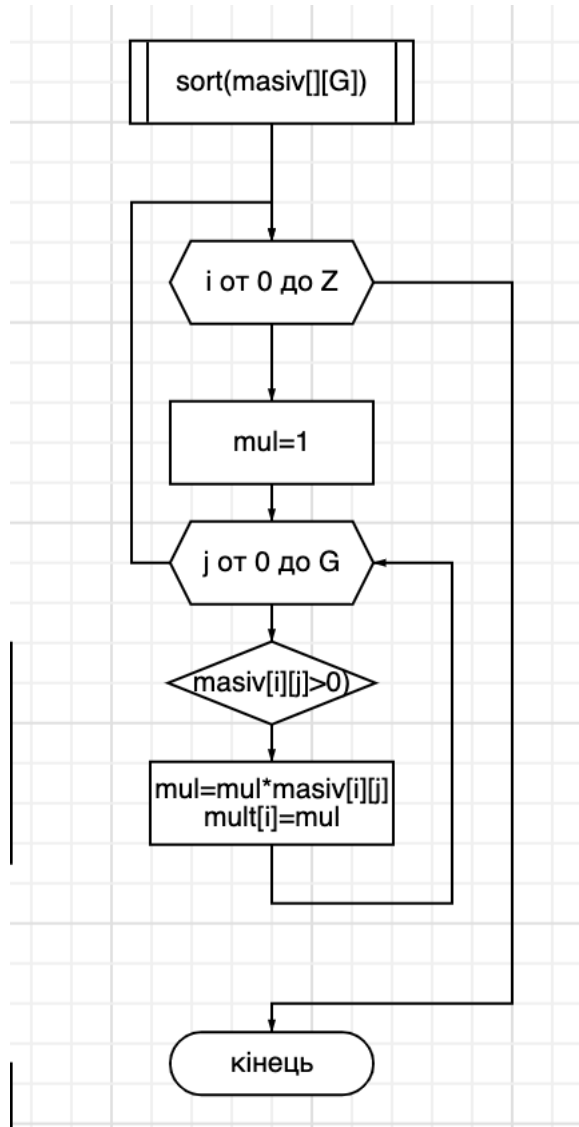
Блок-схема алгоритму

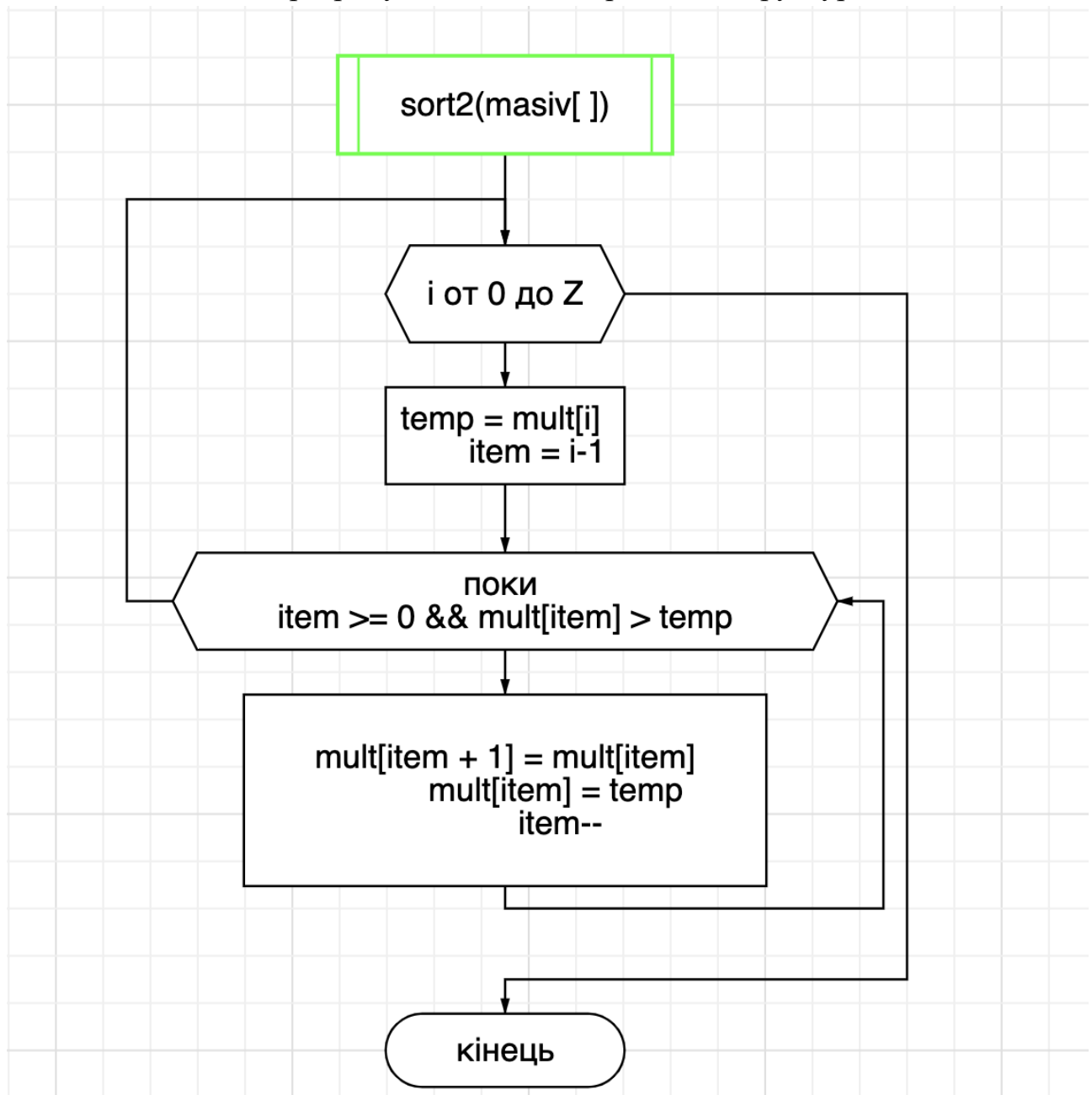


Підпрограми









Код програми

```
1 > #include <iomanip>...
5
6 using namespace std;
7
8 const int Z = 6;
9 const int G = 5;
10
11 int M[Z][G];
12 int mult[Z];
13 int temp,
14     item;
15
16 void input(int[][G]);
17 void output(int[][G]);
18 void sort(int[][G]);
19 void sort2(int[]);
20
21 int main(){
22     srand(time(NULL));
23
24     cout << "Масиви:" << "\n";
25     input(M);
26     output(M);
27     cout << "\n\n\n";
28     sort(M);
29     cout << "\n\n\n";
30     sort2(mult);
31
32     for(int i = 0; i < Z; i++){
33         cout << mult[i] << "    ";
34     }
35     cout << "\n";
36
37     return 0;
38
39 }
40
```

```

42 void input(int masiv[][G]){
43
44     for(int i = 0; i < Z; i++){
45         for(int j = 0; j < G; j++){
46             masiv[i][j] = rand() % 17 - 8;
47         }
48     }
49 }
50
51
52 void output(int masiv[][G]){
53
54     for(int i = 0; i < Z; i++) {
55         cout << "\n";
56         for(int j = 0; j < G; j++){
57             cout << setw(4) << masiv[i][j]  ;
58         }
59     }
60 }
61
62 void sort(int masiv[][G]){
63
64     for(int i = 0; i < Z;i++){
65
66         int mul=1;
67         for(int j = 0; j < G;j++){
68             if(masiv[i][j]>0){
69                 mul=mul*masiv[i][j];
70
71                 mult[i]=mul;
72             }
73         }
74     }
75
76     cout << mult[i] << "  ";
77

```

```

82
83     void sort2(int masiv[]){
84         for (int i = 1; i < Z; i++)
85         {
86             temp = mult[i];
87             item = i-1;
88             while(item >= 0 && mult[item] > temp){
89                 mult[item + 1] = mult[item];
90                 mult[item] = temp;
91                 item--;
92             }
93         }
94     }
95 }
96
97

```

Випробування

```

-5  -8  7  -5  0
0   0  1  -5  2
-4  -6 -3  5  4
8   -5  8  7  6
3   -5  1  -8 -5
8   -1  4  -4  5

7   2   20  2688  3   160

2   3   7   20  160  2688
macbook@Air-MacBook op % cd "/U
Масиви:

6  -6  4  8  5
-7 -5 -7  5  1
-3  6  6  1  4
-2  1  2 -3  7
6   0  5 -1  6
-8  1  0 -7  8

960  5  144  14  180  8

5   8  14  144  180  960

```

Кінець

Висновки

Ми дослідили алгоритми пошуку та сортування, набули практичних навичок використання цих алгоритмів під час складання програмних специфікацій. В результаті виконання лабораторної роботи ми отримали алгоритм для знаходження масиву максимальних значень у рядку матриці та сортування цього масиву.