

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

про виконання лабораторних та практичних робіт блоку № 4

На тему: «Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури.

Алгоритми обробки та робота з масивами та структурами.»

з дисципліни: «Основи програмування»

до:

ВНС Лабораторної Роботи №4

ВНС Лабораторної Роботи №5

Алготестер Лабораторної Роботи №2

Алготестер Лабораторної Роботи №3

Практичних Робіт до блоку №4

Виконав:

Студент групи ШІ-13

Литвин Маркіян Назарович

Львів 2024

Тема роботи: Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами.

Мета: Одержання навичок обробки одновимірних масивів. Організувати обробку масивів з використанням функцій, навчитися передавати масиви як параметри функцій.

Теоретичні відомості:

- масиви
- динамічні масиви
- структури
- вказівники

Джерела:

- <https://www.youtube.com/watch?v=C5yKH8nZTVA>
- <https://www.youtube.com/watch?v=CG69N5MAr88>
- <https://www.youtube.com/watch?v=bQ5me4jAips>
- Chatgpt

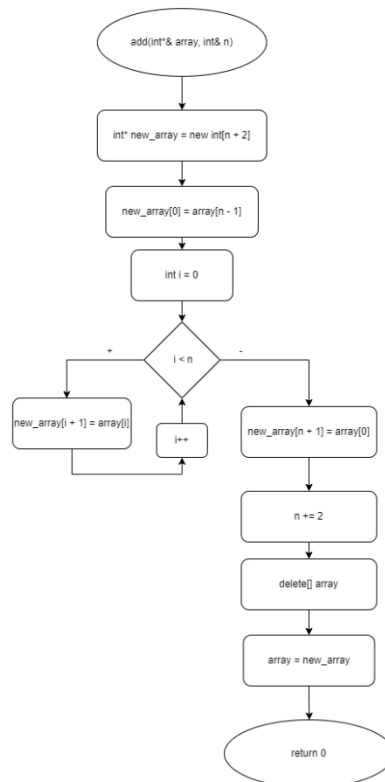
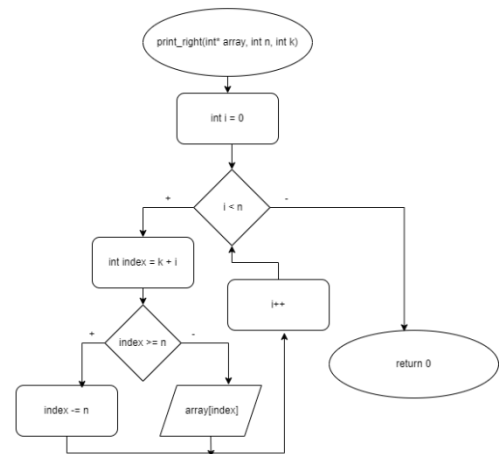
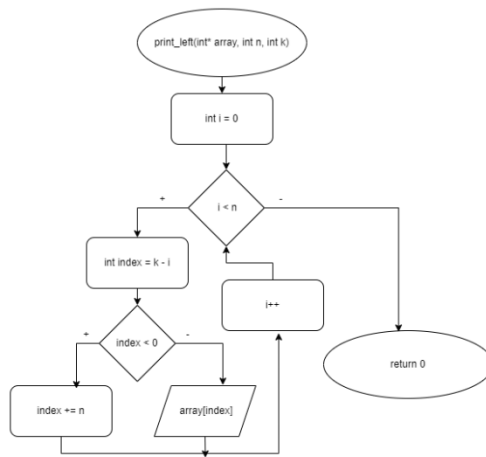
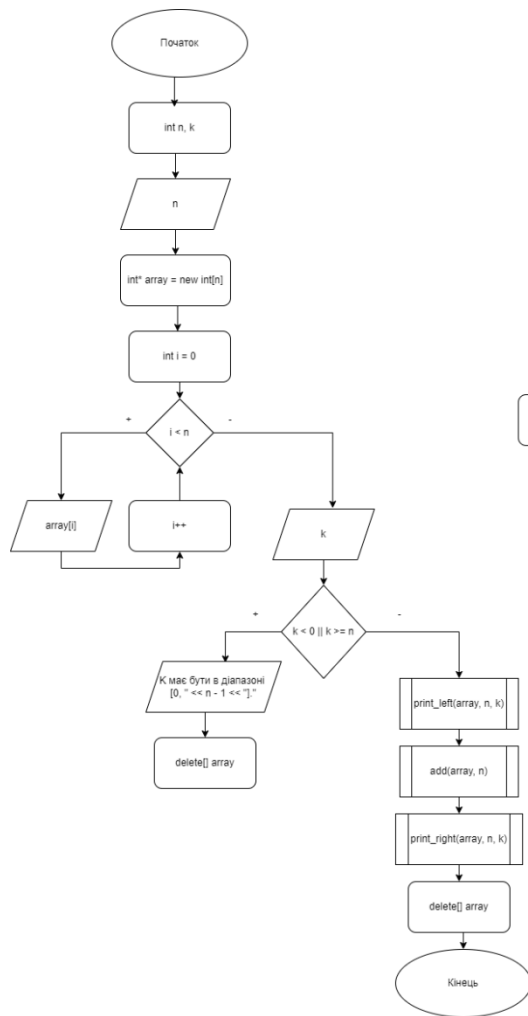
Виконання роботи

Завдання 1: VNS Lab 4 - Task 1-17

Умова:

- 1) Реалізувати з використанням масиву двонаправлене кільце (перегляд можливий в обидва боки, від останнього елемента можна перейти до першого).
- 2) Роздрукувати отриманий масив, починаючи з К-ого елемента й до К-1 (по кільцю вліво).
- 3) Додати в кільце перший і останній елементи.
- 4) Роздрукувати отриманий масив, починаючи з К-ого елемента (і до К+1 по кільцю вправо).

Блок-Схема:



Розв'язок:

```

1  #include <iostream>
2  using namespace std;
3
4  void print_left(int* array, int n, int k) {
5      cout << "Масив по кільцю вліво від K: ";
6      for (int i = 0; i < n; i++) {
7          int index = k - i;
8          if (index < 0){
9              index += n;
10             }
11             cout << array[index] << " ";
12         }
13         cout << endl;
14     }
15
16     void print_right(int* array, int n, int k) {
17         cout << "Масив по кільцю вправо від K: ";
18         for (int i = 0; i < n; i++) {
19             int index = k + i;
20             if(index >= n){
21                 index -= n;
22             }
23             cout << array[index] << " ";
24         }
25         cout << endl;
26     }
27
28     void add(int*& array, int& n) {
29         int* new_array = new int[n + 2];
30         new_array[0] = array[n - 1];
31         for (int i = 0; i < n; i++) {
32             new_array[i + 1] = array[i];
33         }
34         new_array[n + 1] = array[0];
35         n += 2;
36         delete[] array;
37         array = new_array;
38     }
39
40     int main() {
41         int n, k;
42         cout << "Введіть розмір масиву: ";
43         cin >> n;
44
45         if (n <= 0) {
46             cout << "Розмір масиву має бути більшим за 0." << endl;
47             return 0;
48         }
49
50         int* array = new int[n];
51         cout << "Введіть елементи масиву:" << endl;
52         for (int i = 0; i < n; i++) {
53             cin >> array[i];
54         }
55         cout << "Введіть K (індекс стартового елемента): ";
56         cin >> k;
57
58         if (k < 0 || k >= n) {
59             cout << "K має бути в діапазоні [0, " << n - 1 << "]." << endl;
60             delete[] array;
61             return 0;
62         }
63         print_left(array, n, k);
64         add(array, n);
65         print_right(array, n, k);
66         delete[] array;
67         return 0;
68     }
69

```

Результат:

```

Введіть розмір масиву: 3
Введіть елементи масиву:
1 2 3
Введіть K (індекс стартового елемента): 0
Масив по кільцю вліво від K: 1 3 2
Масив по кільцю вправо від K: 3 1 2 3 1

```

Витратив ~ 1.5 год

Завдання 2: VNS Lab 5 - Task 1-17

Умова:

Задано двовимірний масив $N \times M$. Знайти в ньому підмасив 3×3 , сума елементів якого максимальна. N й M можуть бути не кратні трьом.

Розв'язок:

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main(){
6      int N, M;
7      cin >> N >> M;
8      vector<vector<int>> array(N, vector<int>(M));
9      for(int i = 0; i < N; i++){
10         for(int j = 0; j < M; j++){
11             cin >> array[i][j];
12         }
13     }
14     int maxSum = INT_MIN;
15     int r = 0;
16     int c = 0;
17     for (int i = 0; i <= N - 3; i++) {
18         for (int j = 0; j <= M - 3; j++) {
19             int sum = 0;
20             for (int k = i; k < i + 3; k++) {
21                 for (int l = j; l < j + 3; l++) {
22                     sum += array[k][l];
23                 }
24             }
25             if (sum > maxSum) {
26                 maxSum = sum;
27                 r = i;
28                 c = j;
29             }
30         }
31     }
32     cout << endl;
33     cout << "MainMatrix:" << endl;
34     for(int i = 0; i < N; i++){
35         for(int j = 0; j < M; j++){
36             cout << array[i][j] << " ";
37         }
38         cout << endl;
39     }
40     cout << endl;
41     cout << "MaxMatrix 3*3: " << endl;
42     for (int i = r; i < r + 3; i++) {
43         for (int j = c; j < c + 3; j++) {
44             cout << array[i][j] << " ";
45         }
46         cout << endl;
47     }
48     cout << endl;
49     cout << "MaxSum of Matrix 3*3:" << endl;
50     cout << maxSum;
51     return 0;}
```

Результат:

```

4 5
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20

MainMatrix:
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20

MaxMatrix 3*3:
1 2 3
6 7 8
11 12 13

MaxSum of Matrix 3*3:
63

```

Витратив ~ 1 год

Завдання 3: Algotester Lab 2

Умова:

У вас є дорога, яка виглядає як NN чисел.

Після того як ви по ній пройдете - вашу втому можна визначити як різницю максимального та мінімального елемента.

Ви хочете мінімізувати втому, але все що ви можете зробити - викинути одне число з дороги, тобто забрати його з масиву.

В результаті цієї дії, яку мінімальну втому ви можете отримати в кінці дороги?

Input

У першому рядку ціле число NN - кількість чисел

У другому рядку масив rr, який складається з NN цілих чисел

Output

Єдине ціле число mm - мінімальна втома, яку можна отримати

Constraints

$$1 \leq N \leq 10^5, 1 \leq r_i \leq 10^5$$

$$0 \leq r_i \leq 10^5$$

Розв'язок:

```

1  ✓ #include <iostream>
2    #include <vector>
3    #include <algorithm>
4    using namespace std;
5  ✓ int main() {
6      int N;
7      cin >> N;
8      vector<int> r(N);
9  ✓   for (int i = 0; i < N; ++i) {
10     |   cin >> r[i];
11     |   }
12  ✓   if (N <= 2) {
13     |   cout << 0 << endl;
14     |   return 0;
15     |   }
16     sort(r.begin(), r.end());
17     int min_fatigue1 = r[N-1] - r[1];
18     int min_fatigue2 = r[N-2] - r[0];
19  ✓   if (min_fatigue1 > min_fatigue2) {
20     |   cout << min_fatigue2 << endl;
21     |   }
22  ✓   else {
23     |   cout << min_fatigue1 << endl;
24     |   }
25     return 0;
26 }

```

Результат:

```

5
1 3 3 4 4
1
PS D:\Epics>

```

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
2 days ago	C++ 23	Accepted	0.003	1.277	View

Витратив ~ 35 хв

Завдання 4: Algotester Lab 3

Умова:

Вам дано 2 масиви розміром N та M. Значення у цих масивах унікальні.

Ваше завдання вивести у першому рядку кількість елементів, які наявні в обох масивах одночасно, у другому кількість унікальних елементів в обох масивах разом.

Input

У першому рядку ціле число NN

у другому рядку NN цілих чисел a1..ana1..an

У третьому рядку ціле число MM

у четвертому рядку MM цілих чисел $b_1..b_nb_1..b_n$

Output

У першому рядку одне ціле число - кількість елементів, які наявні в обох масивах одночасно.

У другому рядку кількість унікальних елементів в обох масивах (тобто кількість унікальних елементів у масиві, який буде об'єднанням двох даних).

Constraints

$0 \leq N \leq 1000 \leq N \leq 100$

$0 \leq a_i \leq 1000 \leq a_i \leq 100$

$0 \leq M \leq 1000 \leq M \leq 100$

$0 \leq b_i \leq 100$

Розв'язок:

```
1  #include <iostream>
2  #include <vector>
3  #include <set>
4  using namespace std;
5
6  int main() {
7      int N, M;
8      int c = 0;
9      cin >> N;
10     int* arr1=new int[N];
11     for (int i = 0; i < N; i++) {
12         cin >> arr1[i];
13     }
14     cin >> M;
15     int* arr2=new int[M];
16     for (int i = 0; i < M; ++i) {
17         cin >> arr2[i];
18     }
19     for(int i = 0; i < N; i++){
20         for(int j = 0; j < M;j++){
21             if(arr1[i] == arr2[j]){
22                 c++;
23                 break;
24             }
25         }
26     }
27     cout << c << endl;
28     cout << N + M - c;
29     delete[] arr1;
30     delete[] arr2;
31     return 0;
32 }
```

Результат:


```
5
1 2 3 4 5
5
6 7 8 9 10
0
10
PS D:\Epics>
```

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
2 days ago	C++ 23	Accepted	0.003	1.277	View

Витратив ~ 30 хв

Завдання 5: Class Practice Work

Умова:

Реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.

Паліндром — це послідовність символів, яка однаково читається вперед і назад (наприклад, «radar», «level», «12321»).

Мета Задачі

Навчитися користуватися механізмами перевантаження функції та використовувати рекурсію для вирішення задач обчислення.

Вимоги:

- Визначення функції:
 - Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
- Приклад визначення функції:
 - bool isPalindrome(const string& str, int start, int end);*
- Перевантаження функцій:
 - Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.
 - bool isPalindrome(ціле число);*
- Рекурсія:
 - Рекурсивна функція для рядків перевірить символи в поточній початковій і кінцевій позиціях. Якщо вони збігаються, він буде рекурсивно перевіряти наступні позиції, поки початок не перевищить кінець, після чого рядок буде визначено як паліндром.

Розв'язок:

```

1  #include <iostream>
2  #include <string>
3  #include <cmath>
4  using namespace std;
5
6  bool isPalindrome(const string& str, int start, int end) {
7      if (start >= end) {
8          return true;
9      }
10     if (str[start] == str[end]) {
11         return isPalindrome(str, ++start, --end);
12     }
13     return false;
14 }
15 bool isPalindrome(int number) {
16     int invers = 0;
17     int orig = number;
18     while (number > 0) {
19         invers = invers * 10 + number % 10;
20         number /= 10;
21     }
22     return orig == invers;
23 }
24 int main() {
25     int number;
26     cout << "Enter num: ";
27     cin >> number;
28     if (isPalindrome(number)) {
29         cout << number << " is palindrome.\n";
30     } else {
31         cout << number << " isn't palindrome.\n";
32     }
33     string str;
34     cout << "Enter word: ";
35     cin >> str;
36     if (isPalindrome(str, 0, str.size() - 1)) {
37         cout << "\"" << str << "\" is palindrome.\n";
38     } else {
39         cout << "\"" << str << "\" isn't palindrome.\n";
40     }
41     return 0;
42 }

```

Результат:

```

Enter num: 121
121 is palindrome.
Enter word: labs
"labs" isn't palindrome.

```

Витратив ~ 1 год

Завдання 6: Self Practice Work

Умова:

Вам дана стрічка s.

Ваше завдання зробити компресію стрічки, тобто якщо якась буква йде більше одного разу підряд у стрічці замінити її на букву + кількість входжень підряд.

Input

У першому рядку стрічка SS

Output

Стрічка ScompressedScompressed

Constraints

$1 \leq |S| \leq 10$

Розв'язок:

```
1
2  #include <iostream>
3  #include <string>
4  using namespace std;
5
6  int main() {
7      string s, Scomp;
8      cin >> s;
9      int count = 1;
10     for (size_t i = 1; i <= s.size(); ++i) {
11         if (i < s.size() && s[i] == s[i - 1]) {
12             ++count;
13         } else {
14             Scomp += s[i - 1];
15             if (count > 1) {
16                 Scomp += to_string(count);
17             }
18             count = 1;
19         }
20     }
21     cout << Scomp;
22     return 0;
23 }
```

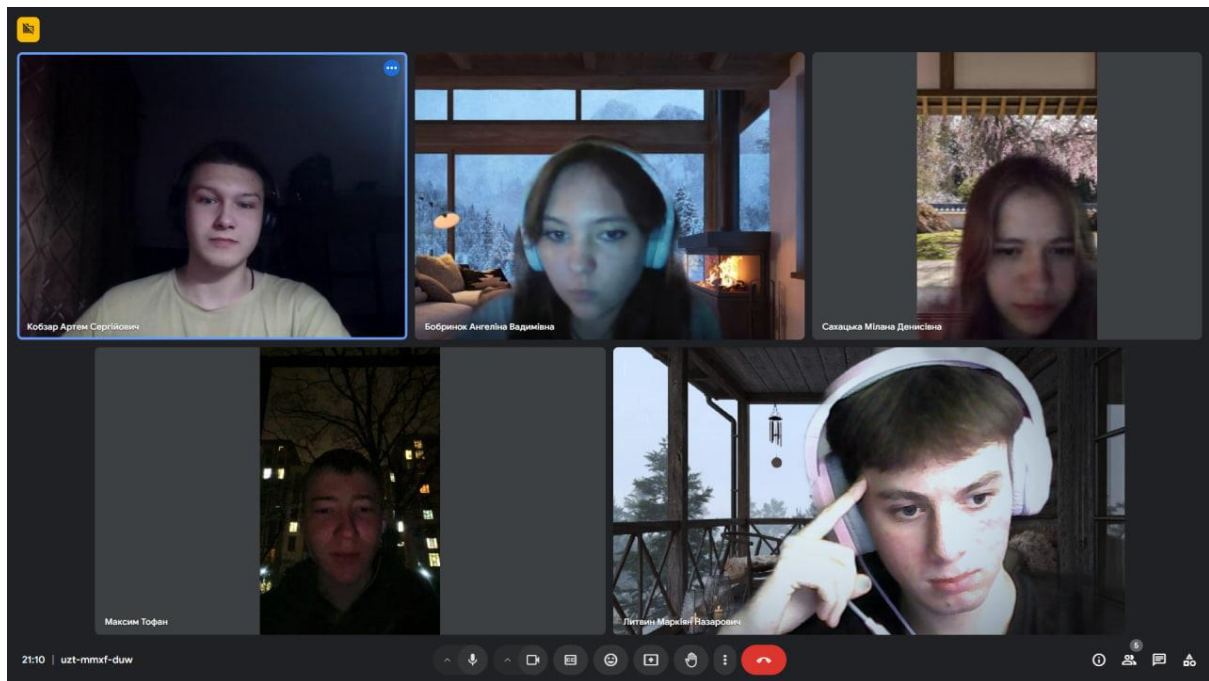
Результат:

```
AAAABBBBCQQQQ
A4B3CQ4
PS D:\Epics> |
```

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
2 days ago	C++ 23	Accepted	0.003	1.215	View

Витратив ~ 30 хв

Зустрічі з командою:



Висновок:

Я отримав навички з обробки одновимірних масивів. Організував обробку масивів з використанням функцій, навчився передавати масиви як параметри функцій.