

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



Звіт

**про виконання розрахунково-графічних робіт блоку № 7
з дисципліни: «Основи програмування»**

до:

ВНС Розрахунково-графічних робіт № 1-4
Практичних Робіт до блоку № 7

Виконав:

Студент групи ІІІ-11
Вербицький Юрій Віталійович

Львів 2024

Тема роботи: Змінні, константи, умовні оператори та розгалуження, масиви, цикли, оператори, функції, робота з файлами, введення та виведення даних, структури, вказівники.

Мета роботи: Узагальнити знання та навички з мов C/C++ здобуті за час курсу «Основ програмування».

Виконання роботи:

1)Перелік завдань:

- Task 1: VNS practice work 1 task 9
- Task 2: VNS practice work 2 task 17
- Task 3: VNS practice work 3 task 14
- Task 4: VNS practice work 4 task 12
- Task 5, 6, 7, 8: Algotester practice work

2)Умови завдань:

Task 1: Розробити лінійний алгоритм для розв'язання задачі.

Варіант 9. $R = x^2(x+1)/b - \sin^2(x-a);$

$S = \sqrt{xb/a} + |\cos(x+b)^3|$, де $a=0,7; b=0,05; x=0,43;$

Task 2: Розробити алгоритм, що розгалужується для розв'язання задачі номер якої відповідає порядковому номеру студента в журналі викладача

Варіант 17. Обчислити функцію двох змінних

$$y = \begin{cases} a^2 + x, & a > x, \\ a^2, & a = x, \text{ де } x \in [1,8]; h_x = 0,5; a \in [2;10]; h_a = 1. \\ a^2 - x, & a < x; \end{cases}$$

Task 3: Написати програму згідно свого варіанту.

Варіант 14. Обчислення площі поверхні циліндра. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення площі поверхні циліндра.

Введіть початкові дані:

Радіус підстави (см) > **5.5**

Висота циліндра (см) > **7**

Площа поверхні циліндра: 431.97 кв.см.

Обчислення опору електричного ланцюга, що складається з двох паралель сполучених резисторів.

Task 4: Написати програму згідно свого варіанту

Варіант 12. Написати програму, яка вводить з клавіатури 5 дробових чисел і обчислює їх середнє арифметичне.

Task 5: Algotester practice Lab1v2

У вас є стіл, у якого є 4 ніжки, довжини яких вам дано.

Ви хочете зробити ніжки рівної довжини, для цього ви відпиляєте dd від кожної ніжки (тобто вам буде дано 4 числа, кожне з яких буде означати відпилювання від відповідної ніжки стола).

Якщо під час відпилювання найдовша ніжка стола буде у 2 рази більша-рівна ніж найменша ніжка - стіл перевернеться, але відпилювати ніжки це вам не завадить. Тобто якщо $h_{max} \geq 2 * h_{min}$ то стіл перевертається. Увага, це може статися і між початком та кінцем відпилювання, наприклад коли відпиляють 2, але ще не встигнуть відпиляти 3тю ніжку. Також ми вважаємо що перед відпилюванням стіл не перевернеться.

Ваше завдання сказати чи після усіх маніпуляцій стіл буде цілий та паралельний підлозі.

Якщо довжина, яку відріжуть буде більша за довжину ножки - вам треба вивести ERROR.

Увага! Навіть якщо стіл перевернеться - ви все одно відпилюєте ніжки і можете отримати ERROR.

Input

4 цілих числа h1,2,3,4- довжини ніжок стола 4 цілих числа d1,2,3,4 - довжина, яку відпиляють від відповідної ножки

Output

YES - якщо стіл буде стояти паралельно площині підлоги та довжина найменшої ніжки не буде рівна нулю.

ERROR - у випадку якщо ви відпиляєте більшу довжину ніж має ножка

NO - у інших випадках

Task 5: Self practice Вам необхідно розробити програму, яка виконує наступні дії:
Зчитує текстовий файл input.txt. Визначає кількість слів у файлі, кількість рядків та кількість символів (включаючи пробіли). Зберігає результати аналізу у файл output.txt у форматі:

Копіювати код

Кількість рядків: X

Кількість слів: Y

Кількість символів: Z

Знаходить усі слова, які починаються з великої літери, і записує їх у файл capitalized_words.txt, кожне слово з нового рядка.

Вхідні дані:

Текстовий файл input.txt, який містить декілька рядків тексту.

Вихідні файли:

Файл output.txt із результатами аналізу.

Файл capitalized_words.txt із усіма словами, що починаються з великої літери.

Task 6: Algotester practice Lab2v1

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 елементу.

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

Task 6: Self practice

Завдання: Аналіз шахової дошки

Розробіть програму для роботи з шаховою дошкою розміром 8x8, яка заповнена значеннями 0 або 1. Програма повинна виконувати наступні дії:

Ввід даних про шахову дошку:

Користувач вводить значення для кожної клітини шахової дошки.

Кожне значення має бути або 0 (порожня клітина), або 1 (заповнена клітина).

Якщо користувач вводить неправильне значення, програма повинна вивести повідомлення про помилку і попросити повторити введення для тієї ж клітини.
Аналіз шахової дошки:

Підрахувати кількість заповнених клітин (1) у кожному рядку.
Визначити координати всіх заповнених клітин (у форматі "рядок, стовець").
Обчислити загальну кількість заповнених клітин на всій дошці.
Вивід результатів:

Для кожного рядка вивести кількість заповнених клітин.
Вивести список координат усіх заповнених клітин.
Вивести загальну кількість заповнених клітин на дошці.

Task 7: Algotester practice Halloween

До Зеника і Марічки на Хелловін завітав їхній старий друг Андрій. Офіційна причина його візиту — дізнатися про справи друзів та просто побалакати з ними. Але Зенику і Марічці добре відомі справжні наміри Андрія — він просто хоче дістати від друзів цукерок.

У Зеника і Марічки є по одному мішку цукерок. Усього в мішку Зеника є n цукерок, причому ii -та із них коштує ai гривень. Аналогічно, у мішку Марічки є m цукерок, і ціна ii -ї рівна bi гривень. Вони хочуть дати Андрію дві цукерки — одну з мішка Зеника, а іншу — з мішка Марічки.

Будучи доволі жадібними, воно готові віддати Андрію лише найдешевші цукерки. Тобто, як і з першого мішка, так і з другого, вони виберуть по найдешевшій цукерці. Якою буде вартість отриманого Андрієм подарунка? Іншими словами, знайдіть сумарну вартість двох цукерок, які отримає Андрій.

Вхідні дані

У першому рядку задано два цілих числа n та m — кількості цукерок в мішках Зеника та Марічки відповідно.

У другому рядку задано n цілих чисел, розділених пробілами. Вони описують вартості Зеникових цукерок.

У третьому рядку аналогічно описані m цукерок Марічки.

Вихідні дані

У єдиному рядку виведіть одне ціле число — вартість подарунку, який отримає Андрій.

Task 7: Self practice

Реалізувати програму для обчислення добутку двох матриць

Task 8: Algotester practice Lab3v3

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 Scompressed

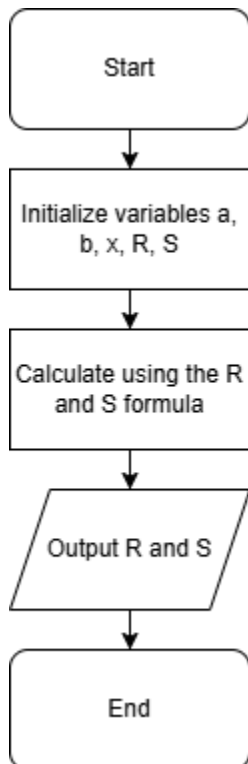
Task 8: Self practice

Розробити програму, яка дозволяє ввести оцінки та обчислити середній бал.

Програма також повинна вивести максимальну і мінімальну оцінку

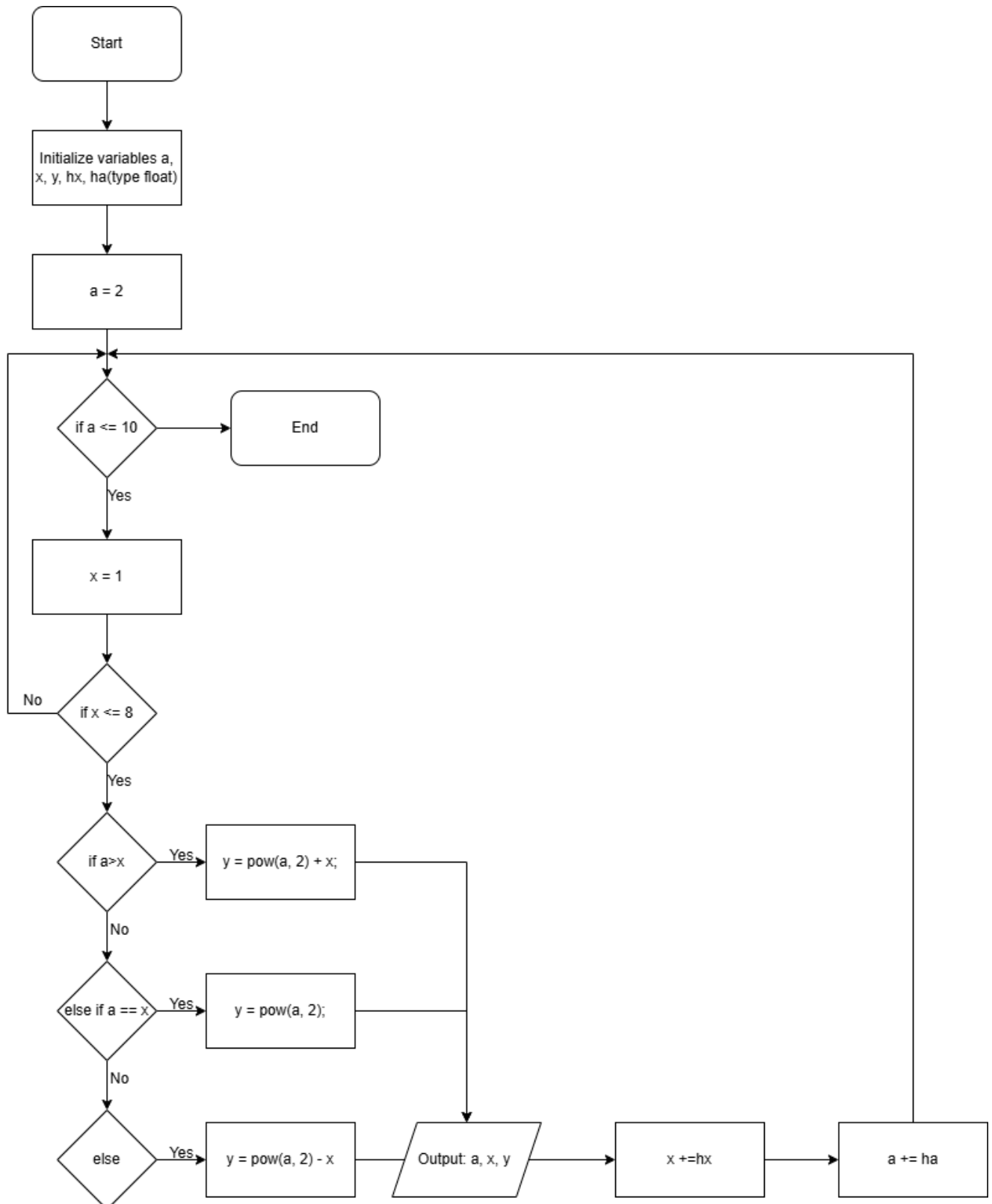
3)Дизайн та планова оцінка часу виконання завдань:

Task 1: VNS practice work 1 task 9



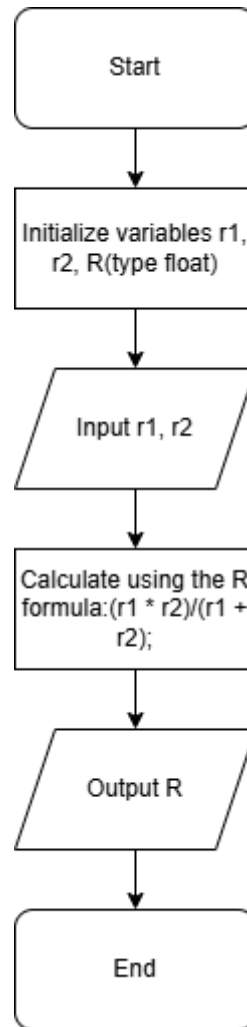
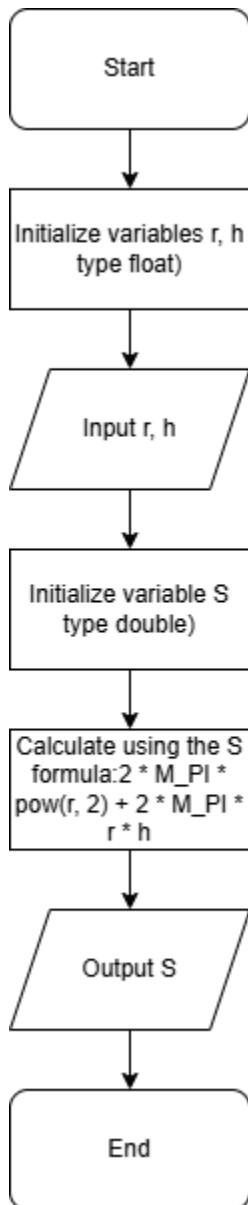
Орієнтовний час виконання: 5хв

Task 2: VNS practice work 2 task 17



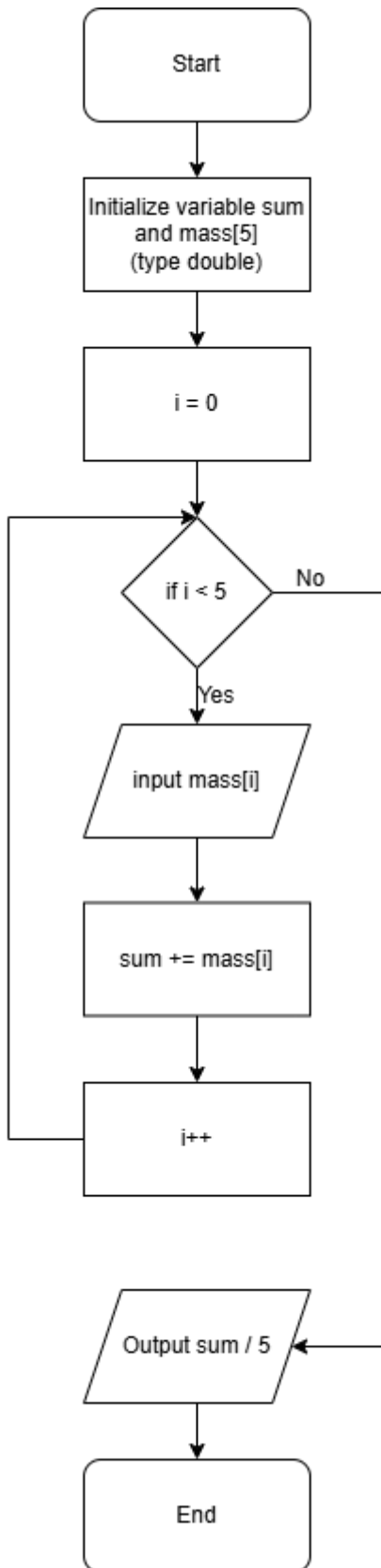
Орієнтовний час виконання: 25хв

Task 3: VNS practice work 3 task 14



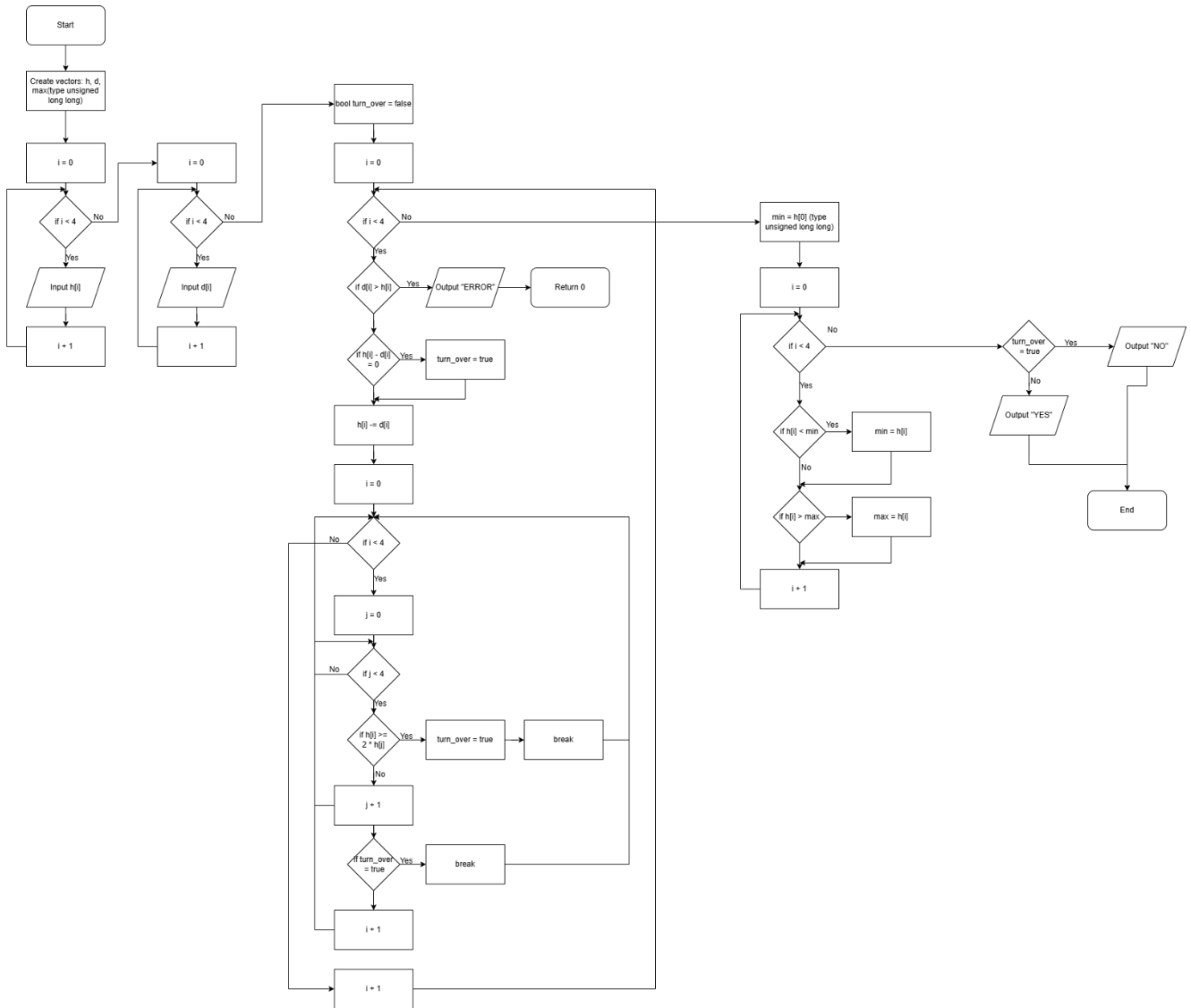
Орієнтовний час виконання: 15хв

Task 4: VNS practice work 4 task 12

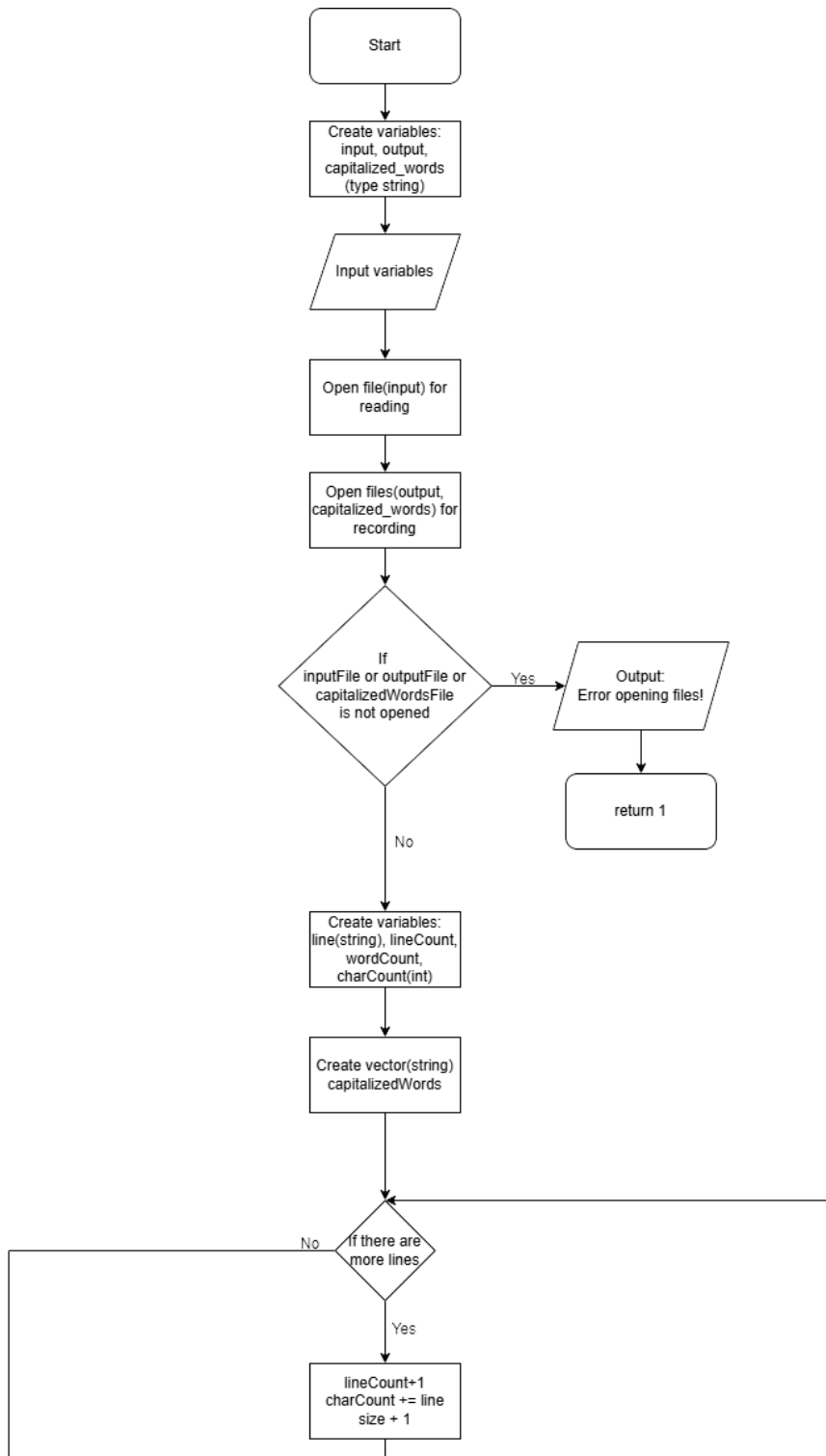


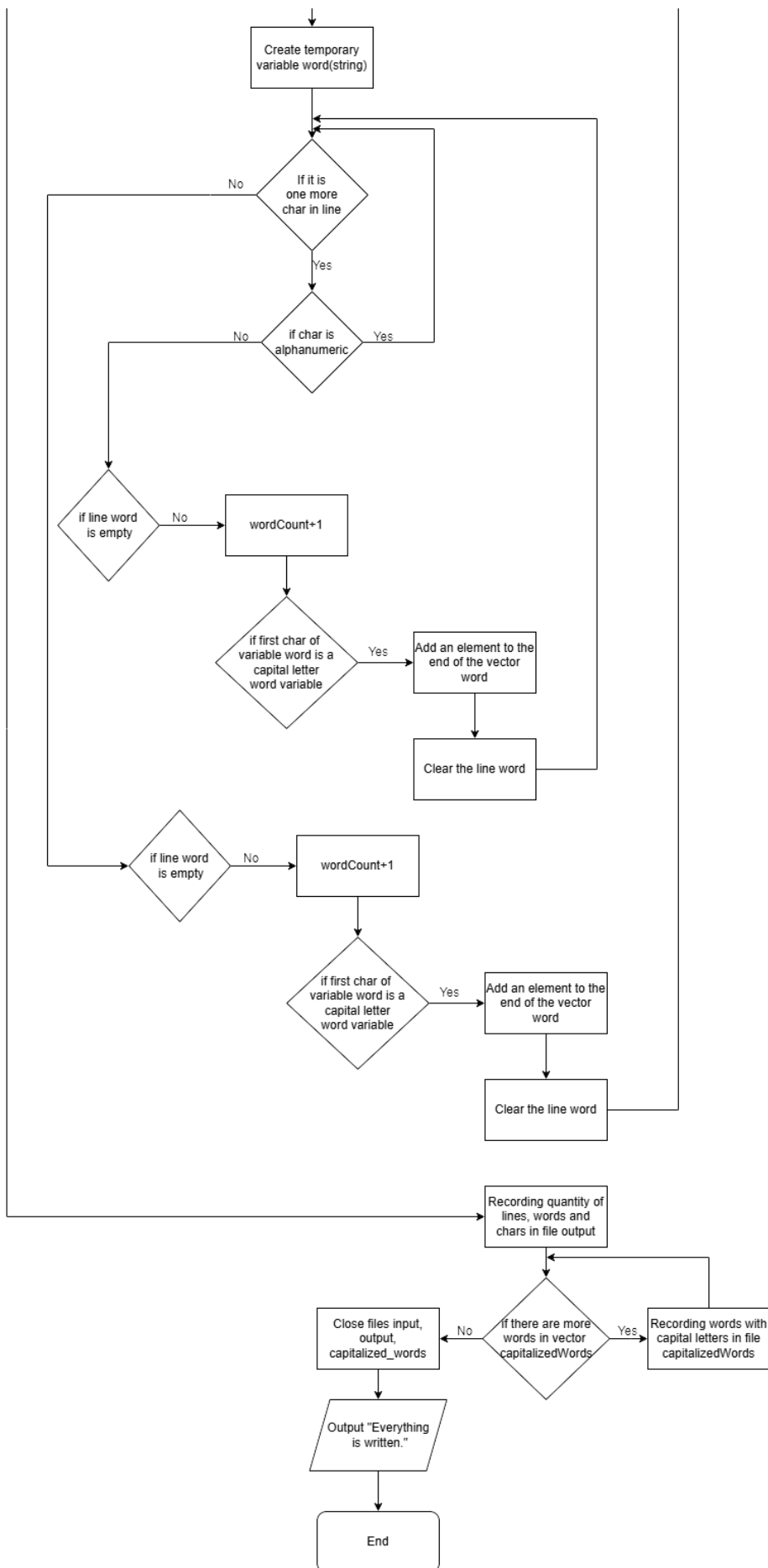
Орієнтовний час виконання: 27хв

Task 5: Algotester practice Lab1v2



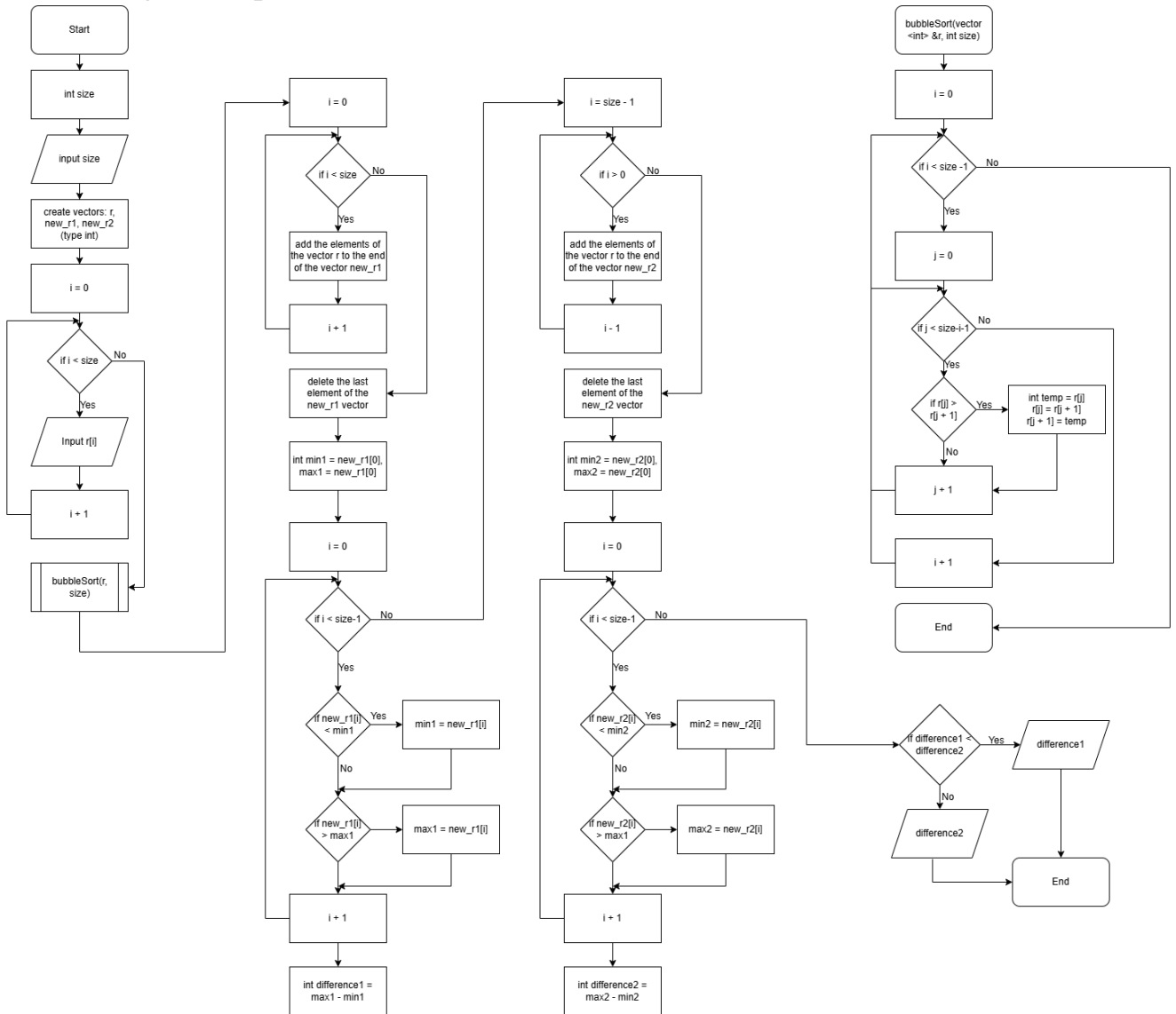
Task 5: Self practice



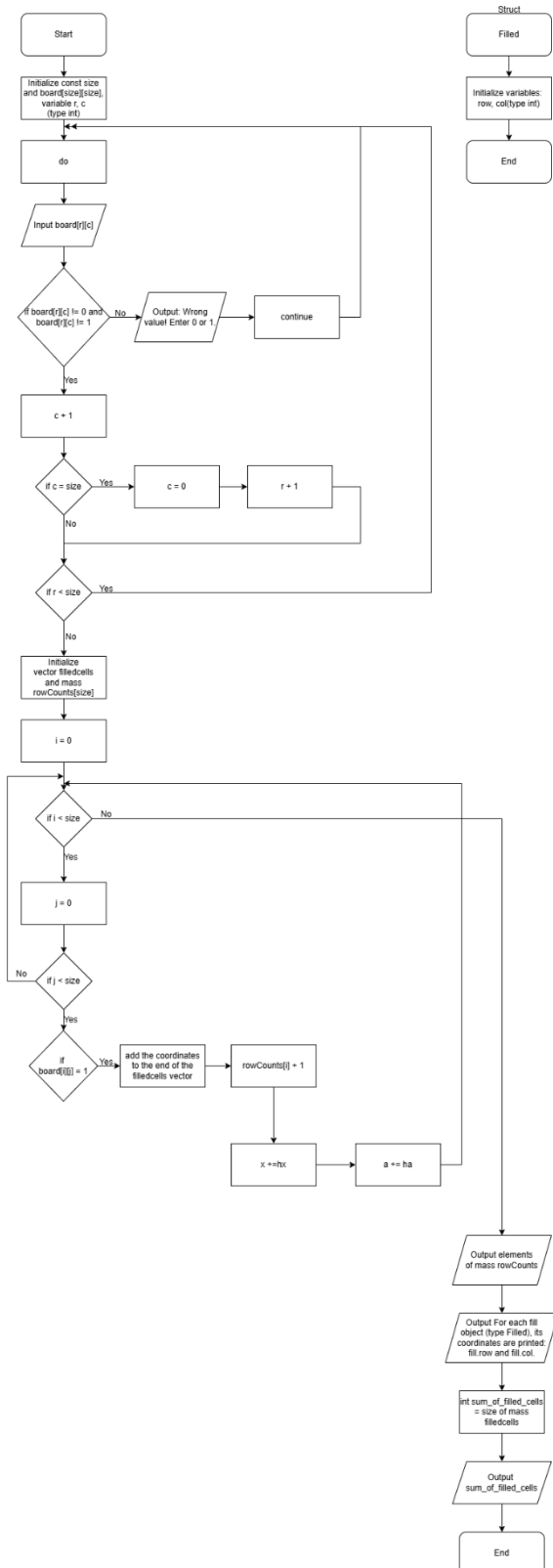


Орієнтовний час виконання: 1 год

Task 6: Algotester practice Lab2v1

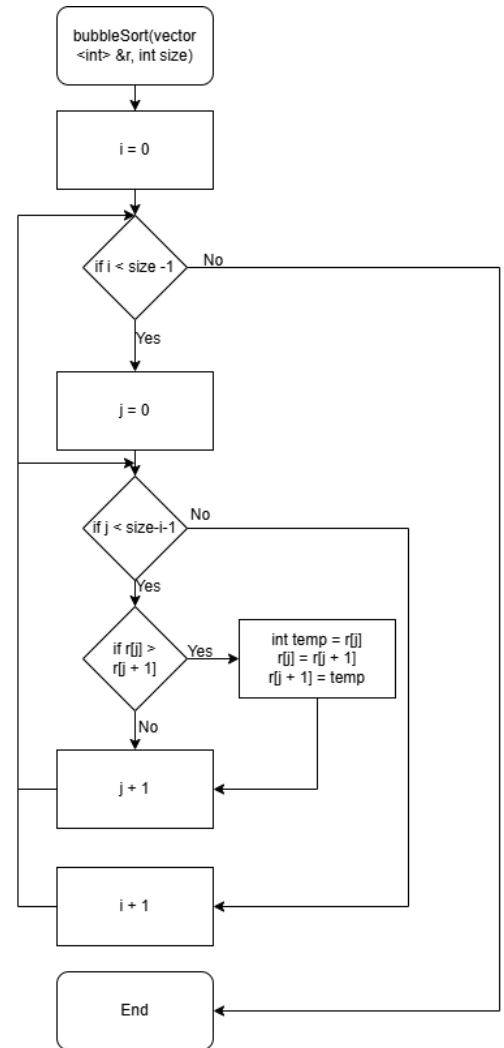
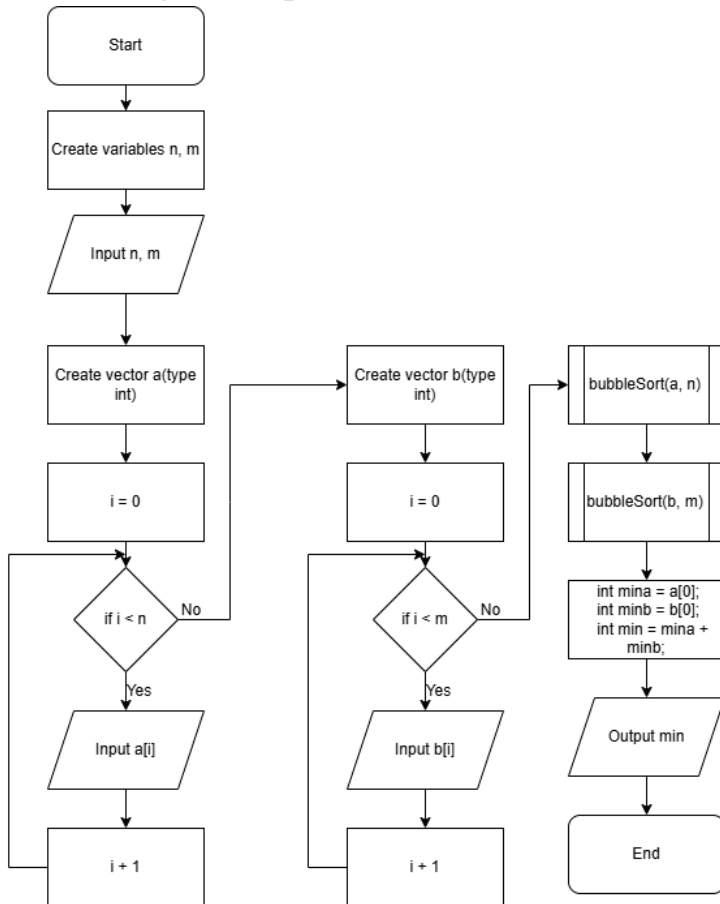


Task 6: Self practice

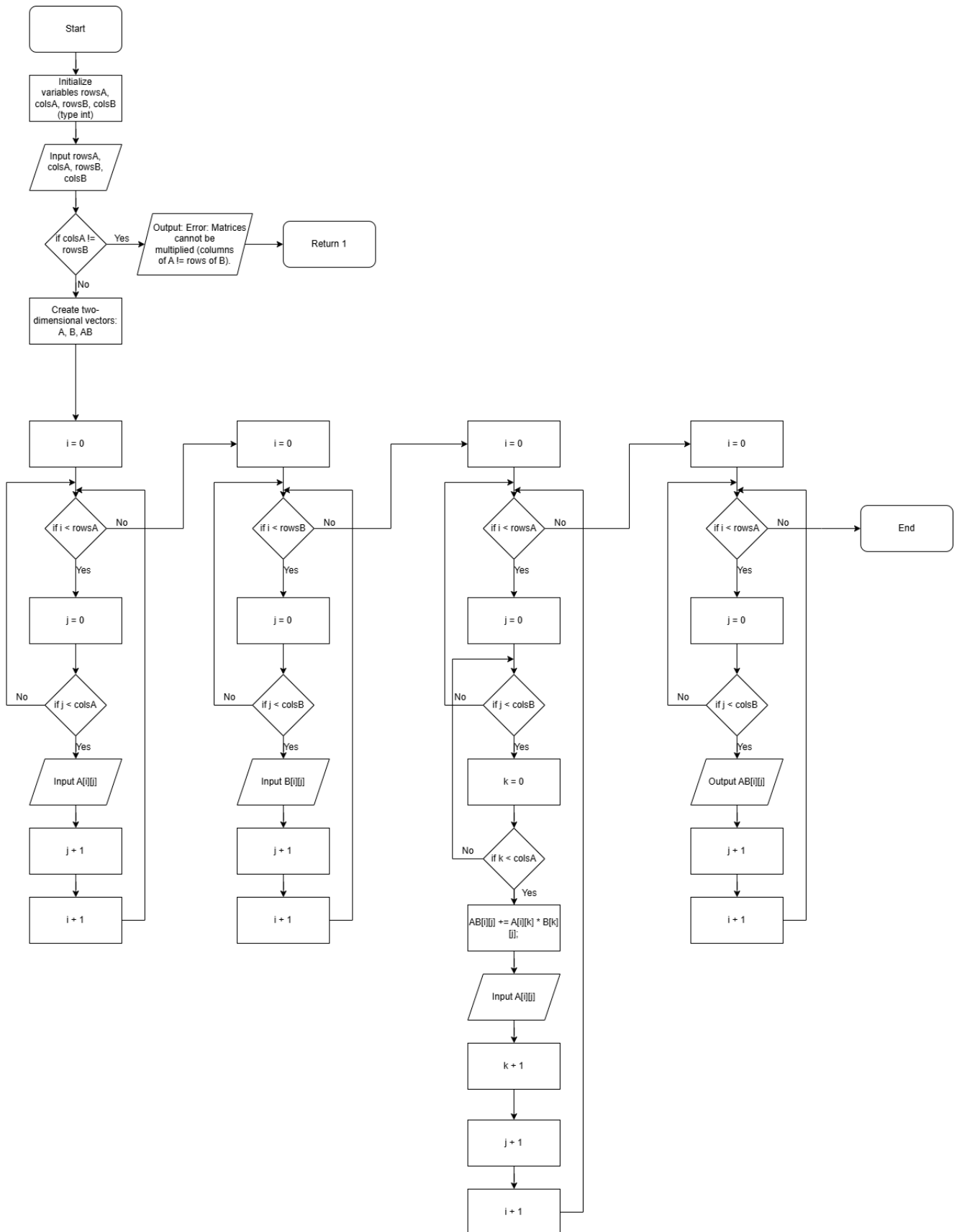


Орієнтовний час виконання: 40хв

Task 7: Algotester practice Halloween

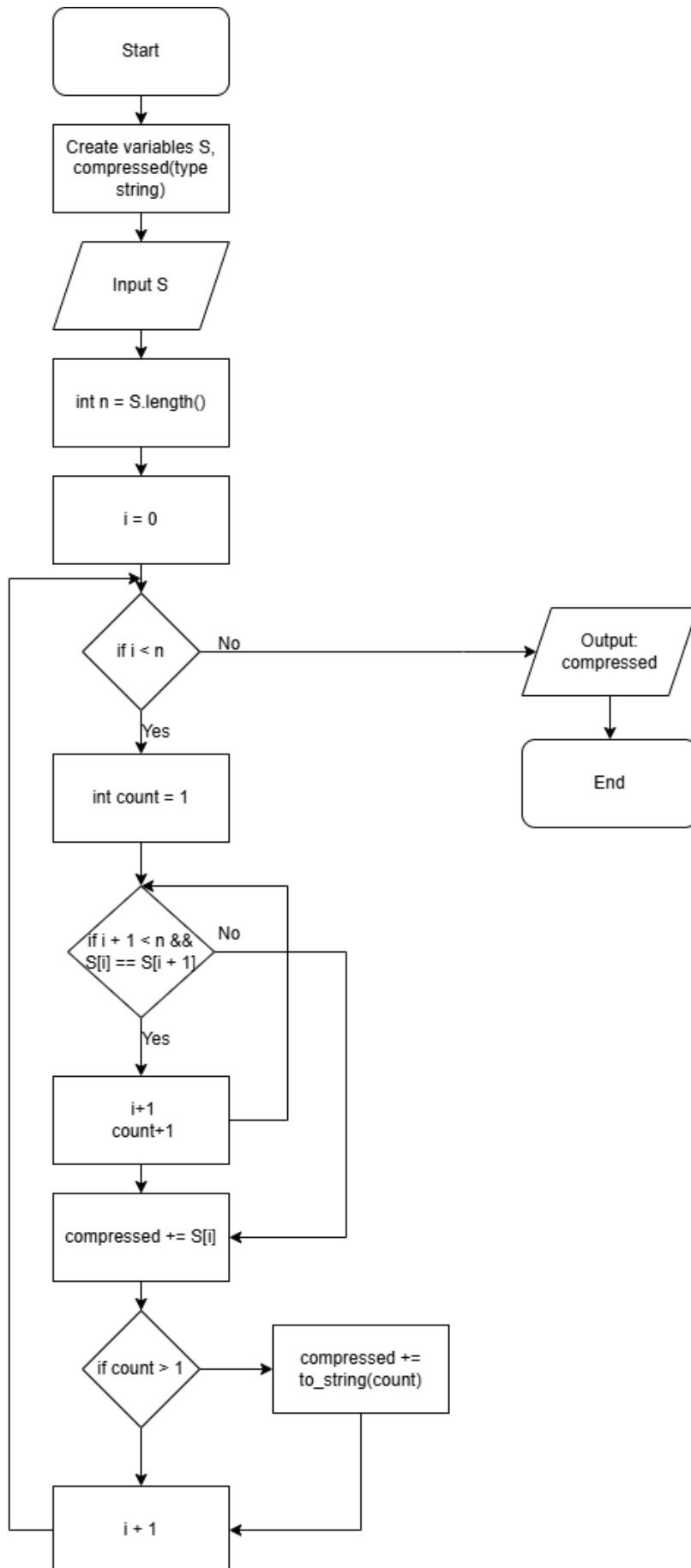


Task 7: Self practice

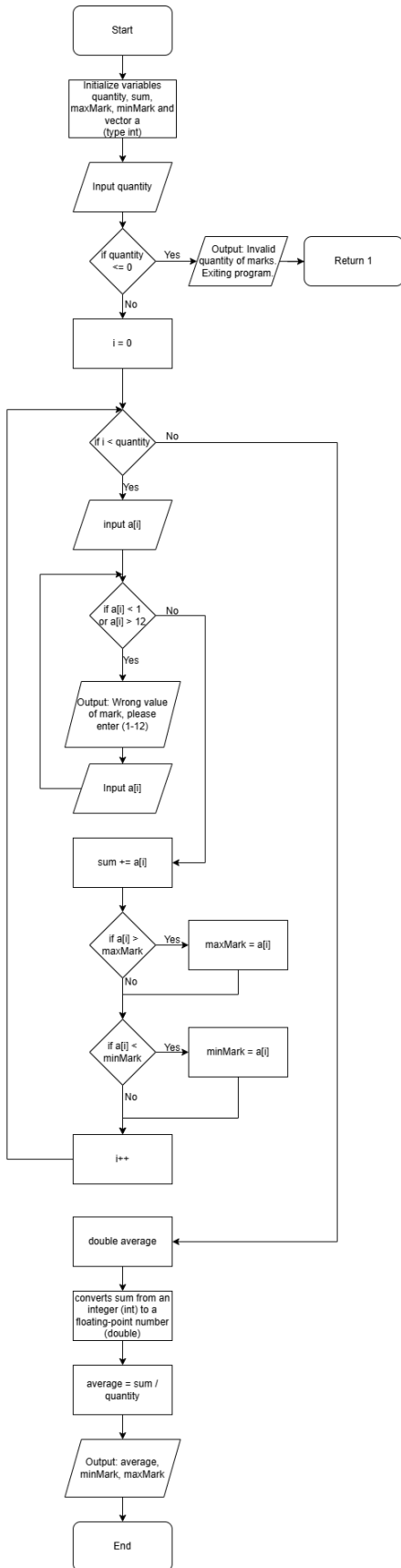


Орієнтовний час виконання: 30хв

Task 8: Algotester practice Lab3v3



Task 8: Self practice



Орієнтовний час виконання: 46хв

4)Код програм з посиланням на зовнішні ресурси:

Task 1: VNS practice work 1 task 9

```
saga_1_vns_practice_work_1_task_yurii_verbytskyi.cpp > ...
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main(){
7      double a = 0.7;
8      double b = 0.05;
9      double x = 0.43;
10     double R; /* використано тип double,
11     тому що числа можуть мати більше чисел після коми, ніж є у float, і передбачити скільки саме буде складно*/
12     double S;
13
14     R = (pow(x, 2) * (x + 1)) / b - pow(sin(x - a), 2);
15     S = sqrt((x * b) / a) + fabs(cos(pow((x + b), 3)));
16     /*використано математичні операції та математичні функції такі як:
17     піднесення до степеня, корінь, модуль, синус та косинус*/
18
19     cout << R << endl;
20     cout << S << endl;
21     /* в кодї використано оператори виведення та введення даних, впринципі як і у всіх наступних кодах,
22     для того, щоб бачити результат виконання програми*/
23
24     return 0;
25 }
```

Task 2: VNS practice work 2 task 17

```
saga_1_vns_practice_work_2_task_yurii_verbytskyi.cpp > ...
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main() {
7      float a, x, y, hx, ha; /*використано тип float,
8      тому що згідно умови ми розуміємо, що жодне із значень не буде мати більше, ніж 1 знак після коми*/
9      hx = 0.5;
10     ha = 1;
11
12
13     for (a = 2; a <= 10; a += ha) {
14         for (x = 1; x <= 8; x += hx) {
15             if (a > x) {
16                 y = pow(a, 2) + x;
17             } else if (a == x) {
18                 y = pow(a, 2);
19             } else {
20                 y = pow(a, 2) - x;
21             }
22             /*в кодї використані умовні оператори та розгалуження для того, щоб обчислити функції з кількома умовами(3)*/
23             cout << "a = " << a << ", x = " << x << ", \t y = " << y << endl;
24         }
25     }
26
27     return 0;
28 }
```

Task 3: VNS practice work 3 task 14

```
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main() {
7      float r, h;
8      cout << "Enter r: ";
9      cin >> r;
10     cout << "Enter h: ";
11     cin >> h;
12     double S;
13     S = 2 * M_PI * pow(r, 2) + 2 * M_PI * r * h;
14     cout << S;
15
16
17     return 0;
18 }
```

```
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main() {
7      float r1, r2;
8      cout << "Enter resistance of the first resistor: ";
9      cin >> r1;
10     cout << "Enter resistance of the second resistor: ";
11     cin >> r2;
12     float R;
13     R = (r1 * r2)/(r1 + r2);
14     cout << "Resistance of electric circuit: " << R;
15
16
17     return 0;
18 }
```

Task 4: VNS practice work 4 task 12

```

1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main() {
7      double mass[5];
8      double sum;
9      for(int i = 0; i < 5; i++){
10         cin >> mass[i];
11         sum += mass[i];
12     }
13     /*в коді використаний for цикл для запису елементів масиву, використаний одновимірний масив
14     для зберігання 5-ти елементів, а також одна цілочисельна змінна(i), щоб обмежити цикл for*/
15     cout << "Arithmetic mean: "<< sum/5;
16 }

```

Task 5: Algotester practice work

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      unsigned long long h[4];
7      for (int i = 0; i < 4; i++) {
8          cin >> h[i];
9      }
10
11     unsigned long long d[4];
12     for (int i = 0; i < 4; i++) {
13         cin >> d[i];
14     }
15
16     unsigned long long max = h[0];
17     bool turn_over = false;
18     for (int i = 0; i < 4; i++) {
19         if (d[i] > h[i]) {
20             cout << "ERROR";
21             return 0;
22         }
23
24         if (h[i] - d[i] == 0) {
25             turn_over = true;
26         }
27
28         h[i] -= d[i];
29         for (int i = 0; i < 4; i++) {
30             for (int j = 0; j < 4; j++) {
31                 if (h[i] >= 2 * h[j]) {
32                     turn_over = true;
33                     break;
34                 }
35             }
36             if (turn_over) {
37                 break;
38             }
39         }
40     }

```

```

40     }
41
42     unsigned long long min = h[0];
43
44     for (int i = 0; i < 4; i++) {
45         if (h[i] < min) {
46             min = h[i];
47         }
48         if (h[i] > max) {
49             max = h[i];
50         }
51     }
52
53     if(turn_over == true){
54         cout << "NO";
55     } else cout << "YES";
56     return 0;
57 }

```

Task 5: Self practice work

```

1  #include <iostream>
2  #include <fstream>
3  #include <sstream>
4  #include <string>
5  #include <cctype>
6  #include <vector>
7  #include <map>
8
9  using namespace std;
10
11  // Files:
12  // input.txt
13  // output.txt
14  // capitalized_words.txt
15
16  int main() {
17      string input, output, capitalized_words;
18
19      cout << "Enter name of input file: ";
20      getline(cin, input);
21      cout << "Enter name of output file: ";
22      getline(cin, output);
23      cout << "Enter name of capitalized words file: ";
24      getline(cin, capitalized_words);
25
26      ifstream inputFile(input);
27      ofstream outputFile(output);
28      ofstream capitalizedWordsFile(capitalized_words);
29
30      if (!inputFile.is_open() || !outputFile.is_open() || !capitalizedWordsFile.is_open()) {
31          cerr << "Error opening files!" << endl;
32          return 1;
33      }
34
35      string line;
36      int lineCount = 0, wordCount = 0, charCount = 0;
37      vector<string> capitalizedWords;
38
39      while (getline(inputFile, line)) {

```

```

40     lineCount++;
41     charCount += line.size() + 1;
42     string word = "";
43
44     for (char ch : line) {
45         if (isalnum(ch)) {
46             word += ch;
47         } else {
48             if (!word.empty()) {
49                 wordCount++;
50
51                 if (isupper(word[0])) {
52                     capitalizedWords.push_back(word);
53                 }
54                 word = "";
55             }
56         }
57     }
58
59     if (!word.empty()) {
60         wordCount++;
61         if (isupper(word[0])) {
62             capitalizedWords.push_back(word);
63         }
64     }
65 }
66
67 outputFile << "Lines: " << lineCount << endl;
68 outputFile << "Words: " << wordCount << endl;
69 outputFile << "Chars: " << charCount << endl;
70
71 for (const string &word : capitalizedWords) {
72     capitalizedWordsFile << word << endl;
73 }
74
75 inputFile.close();
76 outputFile.close();
77 capitalizedWordsFile.close();
78
79 cout << "Everything is written." << endl;
80 return 0;
81 }

```

Task 6: Algotester practice work


```

1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  void bubbleSort(vector<int> &r, int size) {
7      for (int i = 0; i < size - 1; ++i) {
8          for (int j = 0; j < size - i - 1; ++j) {
9              if (r[j] > r[j + 1]) {
10                 int temp = r[j];
11                 r[j] = r[j + 1];
12                 r[j + 1] = temp;
13             }
14         }
15     }
16 }

```

```

17
18 int main() {
19     int size;
20     cin >> size;
21     vector<int> r(size);
22     for(int i = 0; i < size; i++){
23         cin >> r[i];
24     }
25
26     vector<int> new_r1;
27     vector<int> new_r2;
28
29     bubbleSort(r, size);
30
31
32     for (int i = 0; i < size; i++) {
33         new_r1.push_back(r[i]);
34     }
35     new_r1.pop_back();
36     int min1 = new_r1[0], max1 = new_r1[0];
37     for (int i = 0; i < size-1; i++) {
38         if (new_r1[i] < min1) {
39             min1 = new_r1[i];

```

```

40         }
41         if (new_r1[i] > max1) {
42             max1 = new_r1[i];
43         }
44     }
45     int difference1 = max1 - min1;
46
47     for (int i = size - 1; i >= 0; i--) {
48         new_r2.push_back(r[i]);
49     }
50     new_r2.pop_back();
51     int min2 = new_r2[0], max2 = new_r2[0];
52     for (int i = 0; i < size-1; i++) {
53         if (new_r2[i] < min2) {
54             min2 = new_r2[i];
55         }
56         if (new_r2[i] > max2) {
57             max2 = new_r2[i];
58         }
59     }
60     int difference2 = max2 - min2;
61
62     if(difference1 < difference2){
63         cout << difference1;
64     } else cout << difference2;
65
66
67
68     return 0;
69 }

```

Task 6: Self practice work

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  struct Filled {
7      int row;
8      int col;
9  };
10
11  /* Matrix for check
12  0 1 0 0 1 0 0 0
13  0 0 0 0 0 0 1 0
14  1 1 0 0 1 0 0 1
15  0 0 0 0 0 0 0 0
16  1 0 0 1 0 0 1 1
17  0 0 0 0 0 0 0 0
18  0 1 1 0 1 0 0 0
19  0 0 0 1 0 1 0 0
20  */
21  int main() {
22      const int size = 8;
23      int board[size][size];
24
25      int r = 0, c = 0;
26      cout << "Enter board (0 or 1): \n";
27      do {
28          cin >> board[r][c];
29
30          if (board[r][c] != 0 && board[r][c] != 1) {
31              cout << "Wrong value! Enter 0 or 1.\n";
32              continue;
33          }
34          c++;
35          if (c == size) {
36              c = 0;
37              r++;
38          }
39      } while (r < size);
```

```

40
41     vector<Filled> filledcells;
42     int rowCounts[size] = {0};
43
44     for (int i = 0; i < size; i++) {
45         for (int j = 0; j < size; j++) {
46             if (board[i][j] == 1) {
47                 filledcells.push_back({i + 1, j + 1});
48                 rowCounts[i]++;
49             }
50         }
51     }
52
53
54     cout << "\nQuantity of filled cells in each row:\n";
55     for (int i = 0; i < size; i++) {
56         cout << "Row " << i + 1 << ": " << rowCounts[i] << endl;
57     }
58
59
60     cout << "\nCoordinates of filled cells: \n";
61     for (const auto& fill : filledcells) {
62         cout << "(" << fill.row << ", " << fill.col << ") ";
63     }
64     cout << endl;
65
66
67     int sum_of_filled_cells = filledcells.size();
68     cout << "Sum of filled cells: " << sum_of_filled_cells << endl;
69
70     return 0;
71 }

```

Task 7: Algotester practice work

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  void bubbleSort(vector<int> &r, int n) {
7      for (int i = 0; i < n - 1; ++i) {
8          for (int j = 0; j < n - i - 1; ++j) {
9              if (r[j] > r[j + 1]) {
10                 int temp = r[j];
11                 r[j] = r[j + 1];
12                 r[j + 1] = temp;
13             }
14         }
15     }
16 }
17
18 int main() {
19     int n;
20     cin >> n;
21     int m;
22     cin >> m;
23     vector<int> a(n);
24     for(int i = 0; i < n; i++){
25         cin >> a[i];
26     }
27
28     vector<int> b(m);
29     for(int i = 0; i < m; i++){
30         cin >> b[i];
31     }
32     bubbleSort(a, n);
33     bubbleSort(b, m);
34     int mina = a[0];
35     int minb = b[0];
36     int min = mina + minb;
37     cout << min;
38     return 0;
39 }
```

Task 7: Self practice work

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int main() {
7      int rowsA, colsA, rowsB, colsB;
8
9      cout << "Enter dimensions of matrix A (rows cols): ";
10     cin >> rowsA >> colsA;
11     cout << "Enter dimensions of matrix B (rows cols): ";
12     cin >> rowsB >> colsB;
13
14     if (colsA != rowsB) {
15         cout << "Error: Matrices cannot be multiplied (columns of A != rows of B)." << endl;
16         return 1;
17     }
18
19     vector<vector<int>> A(rowsA, vector<int>(colsA));
20     cout << "Enter elements of matrix A:" << endl;
21     for (int i = 0; i < rowsA; i++) {
22         for (int j = 0; j < colsA; j++) {
23             cin >> A[i][j];
24         }
25     }
26
27
28     vector<vector<int>> B(rowsB, vector<int>(colsB));
29     cout << "Enter elements of matrix B:" << endl;
30     for (int i = 0; i < rowsB; i++) {
31         for (int j = 0; j < colsB; j++) {
32             cin >> B[i][j];
33         }
34     }
35
36     vector<vector<int>> AB(rowsA, vector<int>(colsB, 0));
37     for (int i = 0; i < rowsA; i++) {
38         for (int j = 0; j < colsB; j++) {
39             for (int k = 0; k < colsA; k++) {
40
41                 AB[i][j] += A[i][k] * B[k][j];
42             }
43         }
44     }
45
46     cout << "Resulting matrix AB:" << endl;
47     for (int i = 0; i < rowsA; i++) {
48         for (int j = 0; j < colsB; j++) {
49             cout << AB[i][j] << " ";
50         }
51         cout << endl;
52     }
53
54     return 0;
55 }
```

Task 8: Algotester practice work

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  int main() {
7      string S;
8      cin >> S;
9
10     string compressed = "";
11     int n = S.length();
12
13     for (int i = 0; i < n; i++) {
14         int count = 1;
15         while (i + 1 < n && S[i] == S[i + 1]) {
16             i++;
17             count++;
18         }
19
20         compressed += S[i];
21         if (count > 1) {
22             compressed += to_string(count);
23         }
24     }
25
26     cout << compressed << endl;
27
28     return 0;
29 }
```

Task 8: Self practice work

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main() {
6      int quantity;
7
8      cout << "Enter quantity of marks: ";
9      cin >> quantity;
10
11     if (quantity <= 0) {
12         cout << "Invalid quantity of marks. Exiting program." << endl;
13         return 1;
14     }
15
16     vector<int> a(quantity);
17     int sum = 0;
18     int maxMark = INT_MIN;
19     int minMark = INT_MAX;
20
21     cout << "Enter the marks (1-12):" << endl;
22     for (int i = 0; i < quantity; i++) {
23         cin >> a[i];
24         while (a[i] < 1 || a[i] > 12) {
25             cout << "Wrong value of mark, please enter (1-12): ";
26             cin >> a[i];
27         }
28         sum += a[i];
29         if (a[i] > maxMark) {
30             maxMark = a[i];
31         }
32         if (a[i] < minMark) {
33             minMark = a[i];
34         }
35     }
36     double average = static_cast<double>(sum) / quantity;
37     cout << "\nAverage mark: " << average << endl;
38     cout << "Max mark: " << maxMark << endl;
39     cout << "Min mark: " << minMark << endl;
40
41     return 0;
42 }
```

5) Результати виконання завдань та фактично затрачений час

Task 1: VNS practice work 1 task 9

```
5.21699
1.16915
```

Фактично затрачений час: 7хв

Task 2: VNS practice work 2 task 17

| | | |
|-----------------|--------|----------|
| a = 2, x = 1, | y = 5 | |
| a = 2, x = 1.5, | | y = 5.5 |
| a = 2, x = 2, | y = 4 | |
| a = 2, x = 2.5, | | y = 1.5 |
| a = 2, x = 3, | y = 1 | |
| a = 2, x = 3.5, | | y = 0.5 |
| a = 2, x = 4, | y = 0 | |
| a = 2, x = 4.5, | | y = -0.5 |
| a = 2, x = 5, | y = -1 | |
| a = 2, x = 5.5, | | y = -1.5 |
| a = 2, x = 6, | y = -2 | |
| a = 2, x = 6.5, | | y = -2.5 |
| a = 2, x = 7, | y = -3 | |
| a = 2, x = 7.5, | | y = -3.5 |
| a = 2, x = 8, | y = -4 | |
| a = 3, x = 1, | y = 10 | |
| a = 3, x = 1.5, | | y = 10.5 |
| a = 3, x = 2, | y = 11 | |
| a = 3, x = 2.5, | | y = 11.5 |
| a = 3, x = 3, | y = 9 | |
| a = 3, x = 3.5, | | y = 5.5 |
| a = 3, x = 4, | y = 5 | |
| a = 3, x = 4.5, | | y = 4.5 |
| a = 3, x = 5, | y = 4 | |
| a = 3, x = 5.5, | | y = 3.5 |
| a = 3, x = 6, | y = 3 | |
| a = 3, x = 6.5, | | y = 2.5 |
| a = 3, x = 7, | y = 2 | |
| a = 3, x = 7.5, | | y = 1.5 |
| a = 3, x = 8, | y = 1 | |
| a = 4, x = 1, | y = 17 | |
| a = 4, x = 1.5, | | y = 17.5 |
| a = 4, x = 2, | y = 18 | |
| a = 4, x = 2.5, | | y = 18.5 |
| a = 4, x = 3, | y = 19 | |
| a = 4, x = 3.5, | | y = 19.5 |
| a = 4, x = 4, | y = 16 | |
| a = 4, x = 4.5, | | y = 11.5 |
| a = 4, x = 5, | y = 11 | |
| a = 4, x = 5.5, | | y = 10.5 |
| a = 4, x = 6, | y = 10 | |
| a = 4, x = 6.5, | | y = 9.5 |
| a = 4, x = 7, | y = 9 | |

| | | |
|-----------------|--------|----------|
| a = 4, x = 7, | y = 9 | |
| a = 4, x = 7.5, | | y = 8.5 |
| a = 4, x = 8, | y = 8 | |
| a = 5, x = 1, | y = 26 | |
| a = 5, x = 1.5, | | y = 26.5 |
| a = 5, x = 2, | y = 27 | |
| a = 5, x = 2.5, | | y = 27.5 |
| a = 5, x = 3, | y = 28 | |
| a = 5, x = 3.5, | | y = 28.5 |
| a = 5, x = 4, | y = 29 | |
| a = 5, x = 4.5, | | y = 29.5 |
| a = 5, x = 5, | y = 25 | |
| a = 5, x = 5.5, | | y = 19.5 |
| a = 5, x = 6, | y = 19 | |
| a = 5, x = 6.5, | | y = 18.5 |
| a = 5, x = 7, | y = 18 | |
| a = 5, x = 7.5, | | y = 17.5 |
| a = 5, x = 8, | y = 17 | |
| a = 6, x = 1, | y = 37 | |
| a = 6, x = 1.5, | | y = 37.5 |
| a = 6, x = 2, | y = 38 | |
| a = 6, x = 2.5, | | y = 38.5 |
| a = 6, x = 3, | y = 39 | |
| a = 6, x = 3.5, | | y = 39.5 |
| a = 6, x = 4, | y = 40 | |
| a = 6, x = 4.5, | | y = 40.5 |
| a = 6, x = 5, | y = 41 | |
| a = 6, x = 5.5, | | y = 41.5 |
| a = 6, x = 6, | y = 36 | |
| a = 6, x = 6.5, | | y = 29.5 |
| a = 6, x = 7, | y = 29 | |
| a = 6, x = 7.5, | | y = 28.5 |
| a = 6, x = 8, | y = 28 | |
| a = 7, x = 1, | y = 50 | |
| a = 7, x = 1.5, | | y = 50.5 |
| a = 7, x = 2, | y = 51 | |
| a = 7, x = 2.5, | | y = 51.5 |
| a = 7, x = 3, | y = 52 | |
| a = 7, x = 3.5, | | y = 52.5 |
| a = 7, x = 4, | y = 53 | |
| a = 7, x = 4.5, | | y = 53.5 |
| a = 7, x = 5, | y = 54 | |
| a = 7, x = 5.5, | | y = 54.5 |

| | |
|------------------|-----------|
| a = 7, x = 6, | y = 55 |
| a = 7, x = 6.5, | y = 55.5 |
| a = 7, x = 7, | y = 49 |
| a = 7, x = 7.5, | y = 41.5 |
| a = 7, x = 8, | y = 41 |
| a = 8, x = 1, | y = 65 |
| a = 8, x = 1.5, | y = 65.5 |
| a = 8, x = 2, | y = 66 |
| a = 8, x = 2.5, | y = 66.5 |
| a = 8, x = 3, | y = 67 |
| a = 8, x = 3.5, | y = 67.5 |
| a = 8, x = 4, | y = 68 |
| a = 8, x = 4.5, | y = 68.5 |
| a = 8, x = 5, | y = 69 |
| a = 8, x = 5.5, | y = 69.5 |
| a = 8, x = 6, | y = 70 |
| a = 8, x = 6.5, | y = 70.5 |
| a = 8, x = 7, | y = 71 |
| a = 8, x = 7.5, | y = 71.5 |
| a = 8, x = 8, | y = 64 |
| a = 9, x = 1, | y = 82 |
| a = 9, x = 1.5, | y = 82.5 |
| a = 9, x = 2, | y = 83 |
| a = 9, x = 2.5, | y = 83.5 |
| a = 9, x = 3, | y = 84 |
| a = 9, x = 3.5, | y = 84.5 |
| a = 9, x = 4, | y = 85 |
| a = 9, x = 4.5, | y = 85.5 |
| a = 9, x = 5, | y = 86 |
| a = 9, x = 5.5, | y = 86.5 |
| a = 9, x = 6, | y = 87 |
| a = 9, x = 6.5, | y = 87.5 |
| a = 9, x = 7, | y = 88 |
| a = 9, x = 7.5, | y = 88.5 |
| a = 9, x = 8, | y = 89 |
| a = 10, x = 1, | y = 101 |
| a = 10, x = 1.5, | y = 101.5 |
| a = 10, x = 2, | y = 102 |
| a = 10, x = 2.5, | y = 102.5 |
| a = 10, x = 3, | y = 103 |
| a = 10, x = 3.5, | y = 103.5 |
| a = 10, x = 4, | y = 104 |
| a = 10, x = 4.5, | y = 104.5 |

| | |
|------------------|-----------|
| a = 10, x = 5, | y = 105 |
| a = 10, x = 5.5, | y = 105.5 |
| a = 10, x = 6, | y = 106 |
| a = 10, x = 6.5, | y = 106.5 |
| a = 10, x = 7, | y = 107 |
| a = 10, x = 7.5, | y = 107.5 |
| a = 10, x = 8, | y = 108 |

Фактичний час затратений на виконання: 40хв

Task 3: VNS practice work 3 task 14

```
Enter r: 5      Enter resistance of the first resistor: 5
Enter h: 9      Enter resistance of the second resistor: 4
439.823        Resistance of electric circuit: 2.22222
```

Фактичний час затрачений на виконання: 20хв

Task 4: VNS practice work 4 task 12

```
6 5 4 8 12
Arithmetic mean: 7
```


Фактичний час затрачений на виконання: 5хв

Task 5: Algotester practice work

```
10 10 10 10
5 1 1 1
NO
```

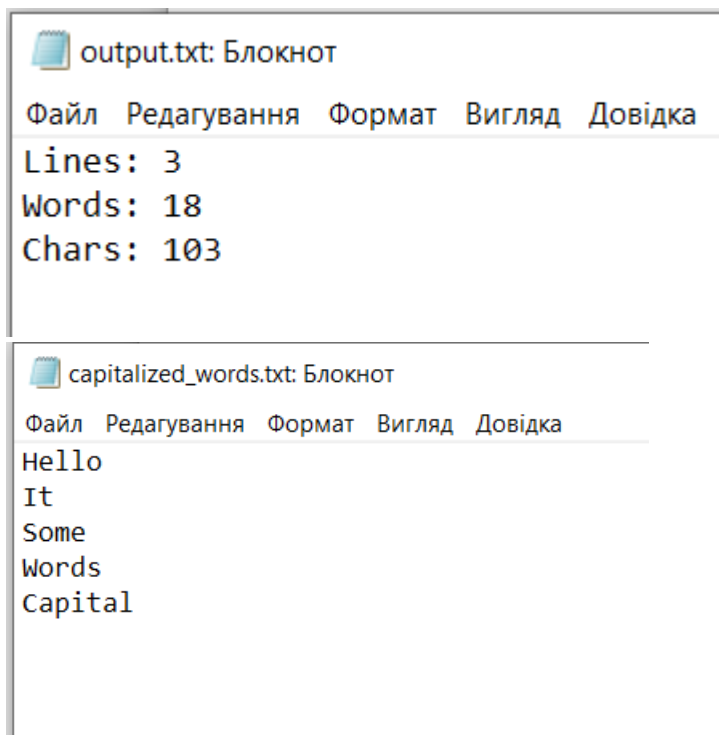
Task 5: Self practice work

```
Enter name of input file: input.txt
Enter name of output file: output.txt
Enter name of capitalized words file: capitalized_words.txt
Everything is written.
PS C:\Users\Yura\Desktop\c++>
```

 input.TXT: Блокнот

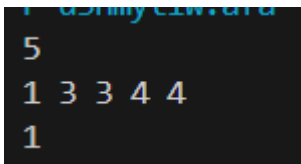
Файл Редагування Формат Вигляд Довідка

```
Hello, this is a sample text file.
It contains multiple lines.
Some words start with a Capital letter.
```

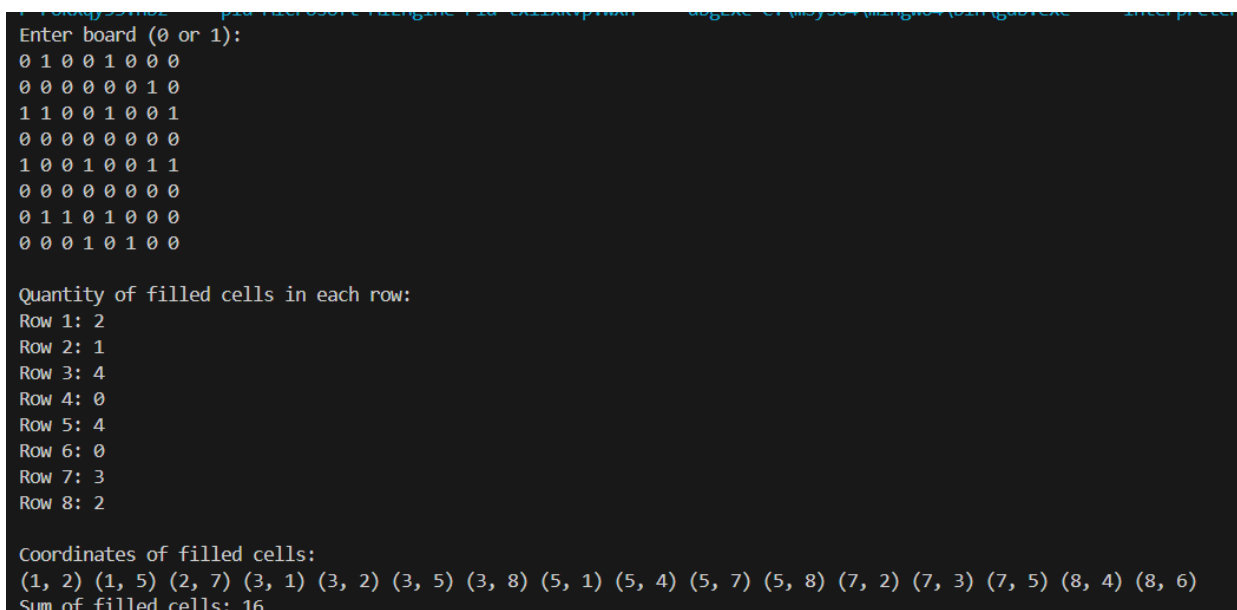


Фактичний час виконання: 1год 29хв

Task 6: Algotester practice work



Task 6: Self practice work



Фактичний час виконання: 25хв

Task 7: Algotester practice work

```
4 7
4 3 2 7
5 2 1 7 6 4 1
3
```

Task 7: Self practice work

```
Enter dimensions of matrix A (rows cols): 3 3
Enter dimensions of matrix B (rows cols): 3 3
Enter elements of matrix A:
1 5 7
7 9 4
3 6 2
Enter elements of matrix B:
12 5 4
1 6 7
2 3 5
Resulting matrix AB:
31 56 74
101 101 111
46 57 64
```

Фактичний час виконання: 40хв

Task 8: Algotester practice work

```
r-2w3wclj.0ep' '--pid=Microso
AAAABBBBCQQQQ
A4B3CQ4
PS C:\Users\Yura\Desktop\c++>
PS C:\Users\Yura\Desktop\c++>
PS C:\Users\Yura\Desktop\c++>
ncher.exe' '--stdin=Microsoft
r-3fdswclj.iu4' '--pid=Microso
AA
A2
PS C:\Users\Yura\Desktop\c++>
PS C:\Users\Yura\Desktop\c++>
PS C:\Users\Yura\Desktop\c++>
ncher.exe' '--stdin=Microsoft
r-w0mkou4v.xkx' '--pid=Microso
A
A
```

Task 8: Self practice work

```
Enter quantity of marks: 10
Enter the marks (1-12):
4 9 7 5 6 3 1 12 4 13
Wrong value of mark, please enter (1-12): 4 9 7 5 6 3 1 12 4 12

Average mark: 5.5
Max mark: 12
Min mark: 1
```

Фактичний час виконання: 27хв

Висновок: Опановуючи роботу з масивами різних типів, вказівниками, посиланнями, динамічними структурами даних та алгоритмами їх обробки, я набув необхідних знань і навичок для ефективного управління даними та їхньої оптимізації. Це дозволяє будувати більш гнучкі та продуктивні програми, здатні працювати з великими обсягами даних, зберігаючи при цьому структуризованість і логічну цілісність інформації.

Посилання на пул реквест: https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/458