

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

Виконала:
Студентка групи ШІ-12
Хвостова Олександра Андріївна

Тема роботи: виконання графічно-розрахункової роботи

Мета роботи: отримати практичні навички в розробці і дослідженні алгоритмів розв'язання

Теоретичні відомості:

- <https://acode.com.ua/uroki-po-cpp/>
- https://www.geeksforgeeks.org/c-plus-plus/?ref=gcse_outind

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

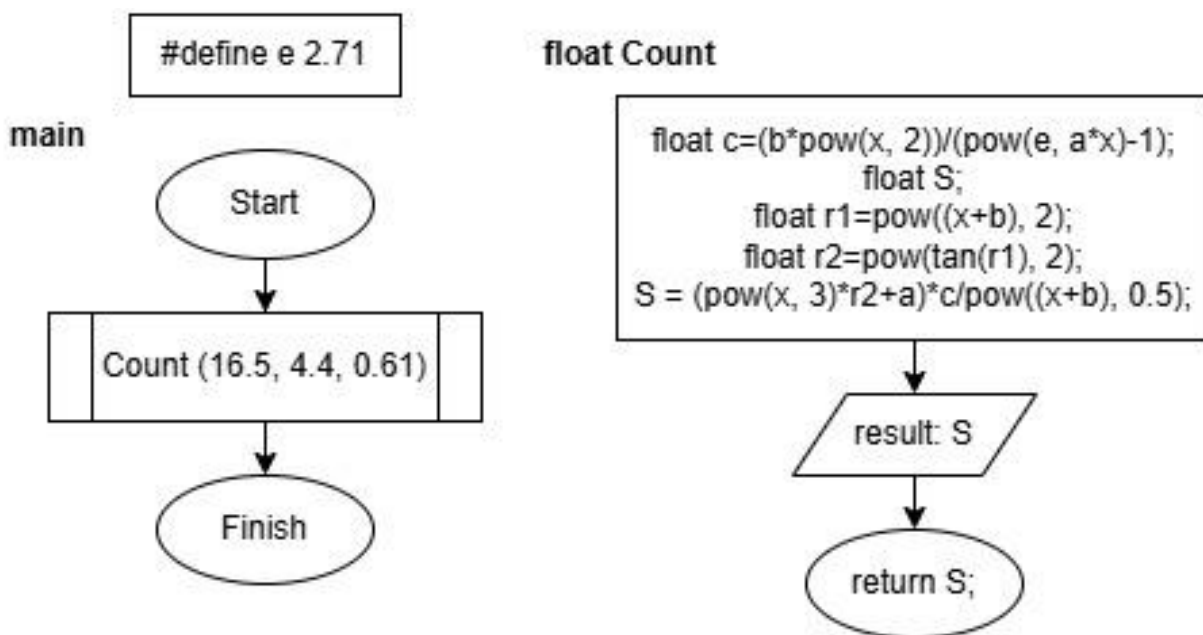
Завдання №1. VNS. Варіант 7

Розробити лінійний алгоритм для розв'язання задачі.

Варіант 7. $S = \frac{x^3 \operatorname{tg}^2(x+b)^2 + a}{\sqrt{x+b}} c$; $c = \frac{bx^2 - c}{e^{ax} - 1}$, де $a=16,5$; $b=3,4$;

$x=0,61$.

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/459/files#diff-f207b5d59380d421ee2cd977d317f4d34f78d7da3dd8cf8fb3a409b23f2ec91b



```

1  #include <math.h>
2  #include <iostream>
3  using namespace std;
4  #define e 2.71
5
6  float Count (float a, float b, float x){
7      float c=(b*pow(x, 2))/(pow(e, a*x)-1);
8      float S;
9      float r1=pow((x+b), 2);
10     float r2=pow(tan(r1), 2);
11     S = (pow(x, 3)*r2+a)*c/pow((x+b), 0.5);
12     cout<<"result: "<<S;
13     return S;
14 }
15
16 int main(){
17     Count(16.5, 3.4, 0.61);
18     return 0;
19 }

```

Результат:

result: 0.000458292

Очікуваний час виконання: 1 година

Реальний час виконання: 30 хвилин

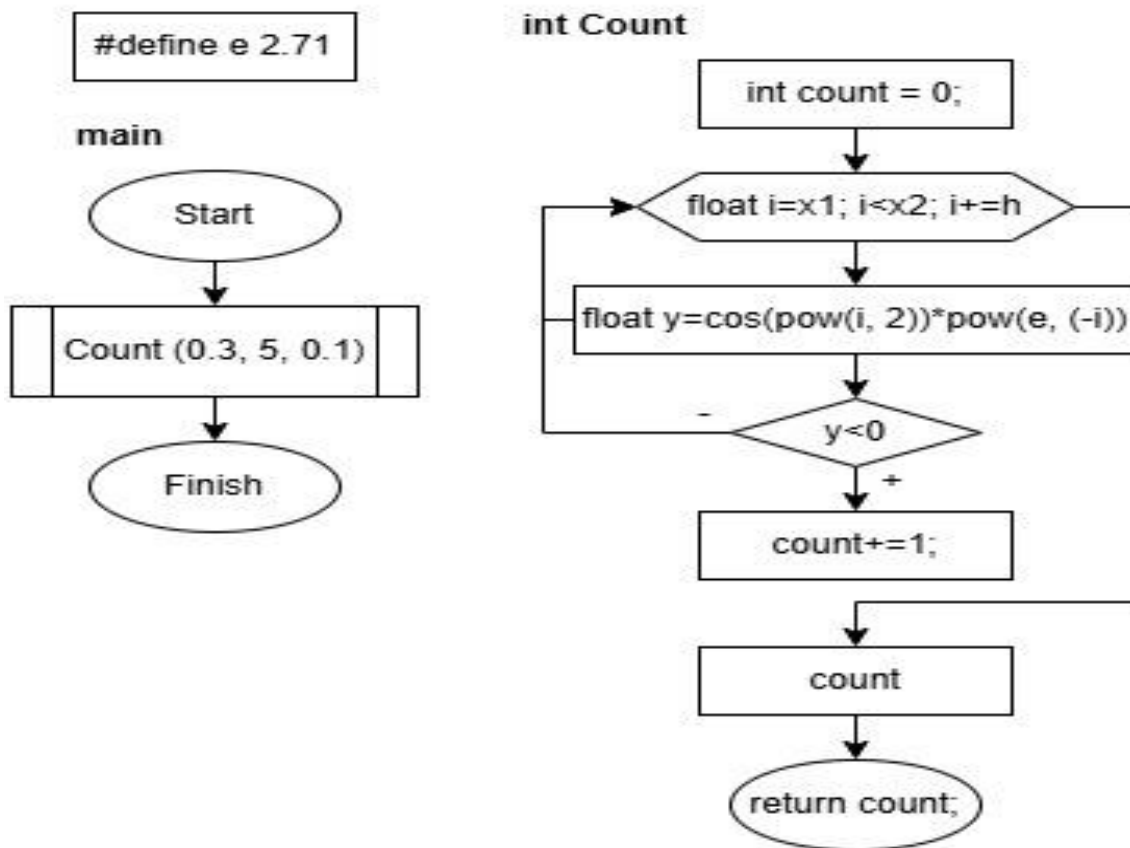
Завдання №2. VNS. Варіант 24

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

Варіант 24. Підрахувати, скільки разів функція $y = \cos x^2 \cdot e^{-x}$

приймає негативне значення, якщо $x \in [0,3;5]$; $h_x = 0,1$.

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/41645432c95a57c2a05b55a5787ab90ee58838c1/ai_12/oleksandra_khvostova/epic_7/vns_practice_work_2_task_oleksandra_khvostova.cpp



```

1  #include <iostream>
2  #include <math.h>
3  using namespace std;
4  #define e 2.71
5
6  int Count (float x1, float x2, float h){
7      int count=0;
8      for(float i=x1; i<=x2; i+=h){
9          float y=cos(pow(i, 2))*pow(e, (-i));
10         if (y<0){
11             count+=1;
12         }
13     }
14     cout<<count;
15     return count;
16 }
17
18 int main(){
19     Count(0.3, 5, 0.1);
20     return 0;
21 }

```

Результат:

21

Очікуваний час виконання: 1 година

Реальний час виконання: 20 хвилин

Завдання №3. VNS. Варіант 19

Варіант 19. Обчислює площу трикутника, якщо відомі координати його кутів. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, введені користувачем, які вводяться напівжирним шрифтом).

Обчислення площі трикутника.

Введіть координати кутів

(числа розділяйте пропуском):

x1,y1 > **-2 5**

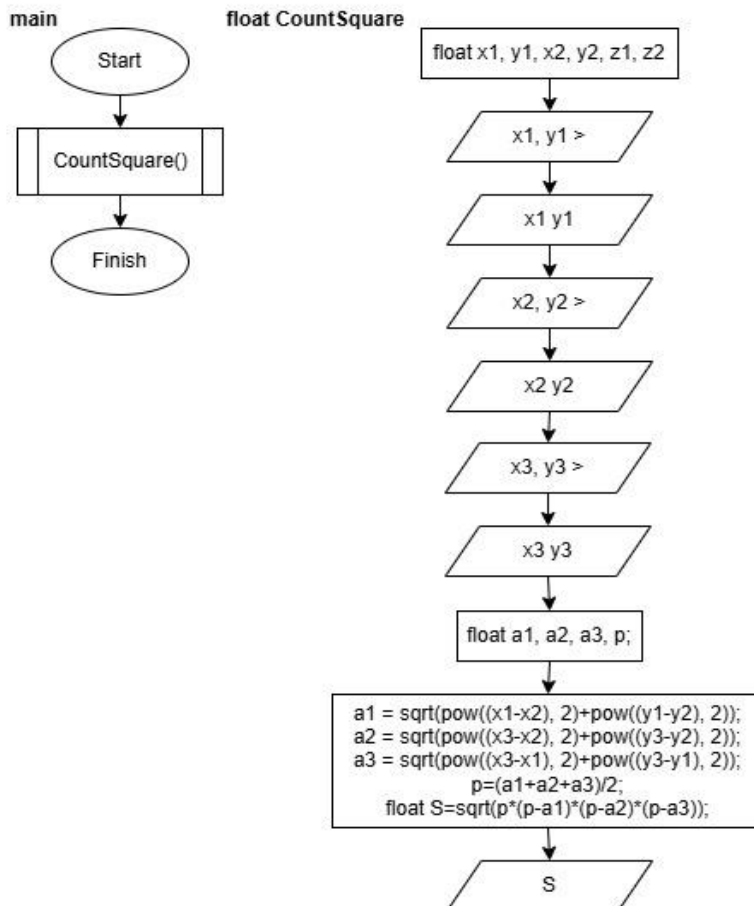
x2,y2 > **1 7**

x3,y3 > **5 -3**

Площа трикутника: 23.56 кв.см.

Обчислення вартості покупки, що складається з набору зошитів і олівців.

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/41645432c95a57c2a05b55a5787ab90ee58838c1/ai_12/oleksandra_khvostova/epic_7/vns_practice_work_3_task_oleksandra_khvostova.cpp



```

1  #include <iostream>
2  #include <math.h>
3  using namespace std;
4
5  float CountSquare(){
6      float x1, y1, x2, y2, x3, y3;
7      cout<<"x1, y1 > ";
8      cin>>x1>>y1;
9      cout<<endl;
10     cout<<"x2, y2 > ";
11     cin>>x2>>y2;
12     cout<<endl;
13     cout<<"x3, y3 > ";
14     cin>>x3>>y3;
15     cout<<endl;
16     float a1, a2, a3, p;
17     a1=sqrt(pow((x1-x2), 2) + pow((y1-y2), 2));
18     a2=sqrt(pow((x3-x2), 2) + pow((y3-y2), 2));
19     a3=sqrt(pow((x3-x1), 2) + pow((y3-y1), 2));
20     p=(a1+a2+a3)/2;
21     float S=sqrt(p*(p-a1)*(p-a2)*(p-a3));
22     cout<<S<<endl;
23     return S;
24 }
25
26 int main(){
27     CountSquare();
28     int N1, N2, V1, V2, res;
29     cout<<"Кількість зошитів у наборі: ";
30     cin>>N1;
31     cout<<"Вартість одного зошита: ";
32     cin>>V1;
33     cout<<"Кількість олівців у наборі: ";
34     cin>>N2;
35     cout<<"Вартість одного олівця: ";
36     cin>>V2;
37     res=N1*V1+N2*V2;
38     cout<<"Вартість набору: "<<res<<endl;
39     return 0;
40 }

```

Результат:

```

x1, y1 > 1 0

x2, y2 > 6 0

x3, y3 > 4 5

12.5
Кількість зошитів у наборі: 4
Вартість одного зошита: 12
Кількість олівців у наборі: 3
Вартість одного олівця: 6
Вартість набору: 66

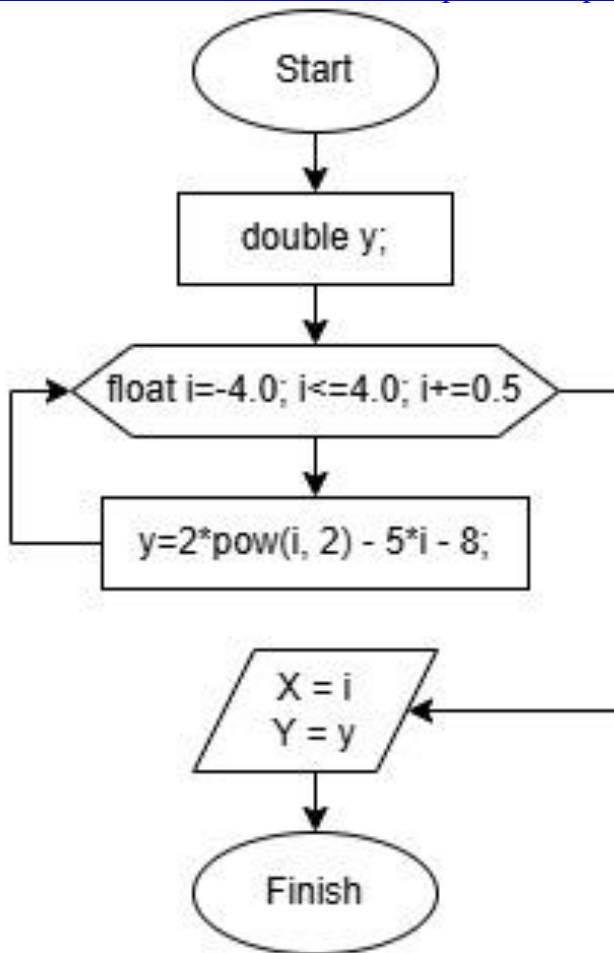
```

Очікуваний час виконання: 1 година

Реальний час виконання: 20 хвилин

Варіант 2. Напишіть програму, яка виводить на екран таблицю значень функції $y = 2x^2 - 5x - 8$ в діапазоні від -4 до 4. Крок зміни аргументу 0,5.

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/41645432c95a57c2a05b55a5787ab90ee58838c1/ai_12/oleksandra_khvostova/epic_7/vns_practice_work_4_task_oleksandra_khvostova.cpp



```
1  #include <iostream>
2  #include <iomanip>
3  #include <math.h>
4  using namespace std;
5
6  int main(){
7      double y;
8      for (float i=-4.0; i<=4.0; i+=0.5){
9          y=2*pow(i, 2)-5*i-8;
10         cout<<"X = "<<i<<setw(4)<<"\t"<<"Y = "<<y<<endl;
11     }
12     return 0;
13 }
```

Результат:

X = -4	Y = 44
X = -3.5	Y = 34
X = -3	Y = 25
X = -2.5	Y = 17
X = -2.5	Y = 17
X = -2	Y = 10
X = -1.5	Y = 4
X = -1	Y = -1
X = -0.5	Y = -5
X = 0	Y = -8
X = 0.5	Y = -10
X = 1	Y = -11
X = 1.5	Y = -11
X = 2	Y = -10
X = 2.5	Y = -8
X = 3	Y = -5
X = 3.5	Y = -1
X = 4	Y = 4

Очікуваний час виконання: 1 година

Реальний час виконання: 20 хвилин

Завдання №5. Algotester. Lab1v3

Lab 1v3

Обмеження: 1 сек., 256 МБ

Персонажу по одному дають сторони 5 кубів $a_{1..5}$, з яких він будує піраміду.

Коли він отримує куб з ребром a_i - він його ставить на існуючий, перший ставить на підлогу (вона безмежна).

Якщо в якийсь момент об'єм куба у руці (який будуть ставити) буде більший ніж у куба на вершині піраміди - персонаж програє і гра закінчується. Розмір усіх наступних кубів після програшу **не враховується**.

Тобто якщо $a_{i-1} < a_i$ - це програш.

Ваше завдання - сказати як закінчиться гра.

Вхідні дані

5 цілих чисел $a_1 \dots a_5$ - сторони кубів

Вихідні дані

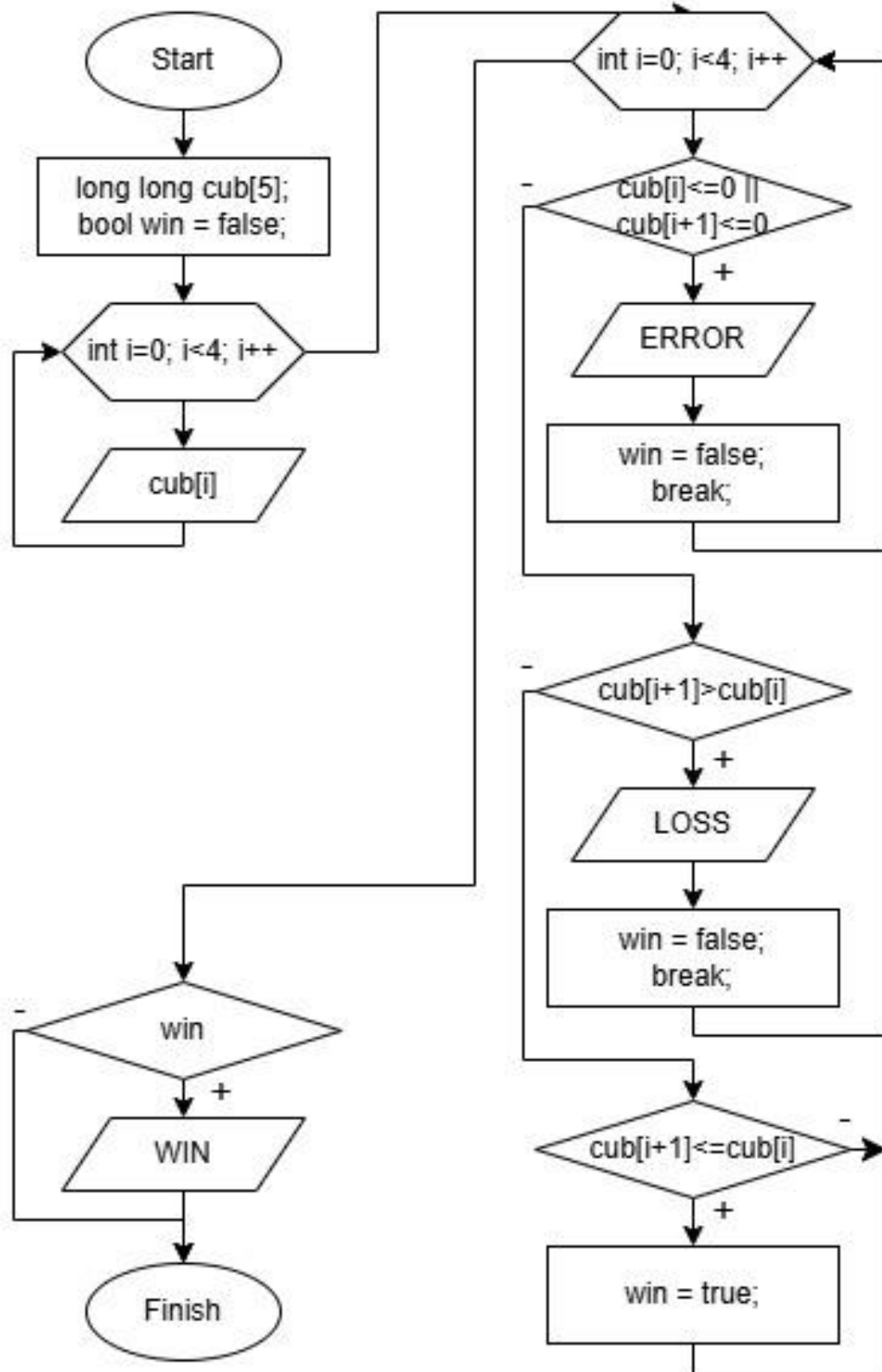
Існуючі варіанти:

LOSS - якщо персонаж не зможе поставити куб.

WIN - якщо персонаж зможе поставити усі куби

ERROR - якщо сторона куба буде мати неможливу довжину, тобто $a_i \leq 0$

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/41645432c95a57c2a05b55a5787ab90ee58838c1/ai_12/oleksandra_khvostova/epic_7/algotester_lab1_v3.cpp



```

1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      long long cub[5];
6      bool win=false;
7      for(int i=0; i<5; i++){
8          cin>>cub[i];
9      }
10     for(int i=0; i<4; i++){
11         if(cub[i]<=0||cub[i+1]<=0){cout<<"ERROR"; win=false; break;}
12         if(cub[i+1]>cub[i]){cout<<"LOSS"; win=false; break;}
13         if(cub[i+1]<=cub[i]){win = true;}
14     }
15     if(win){
16         cout<<"WIN";
17     }
18     return 0;
19 }
20

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)
9 днів тому	C++ 23	Зараховано	0.003	1.313

Очікуваний час виконання: 1 година

Реальний час виконання: 20 хвилин

Завдання №6. Algotester. Lab2v1

Lab 2v1

Обмеження: 1 сек., 256 МіБ

У вас є дорога, яка виглядає як N 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?

Вхідні дані

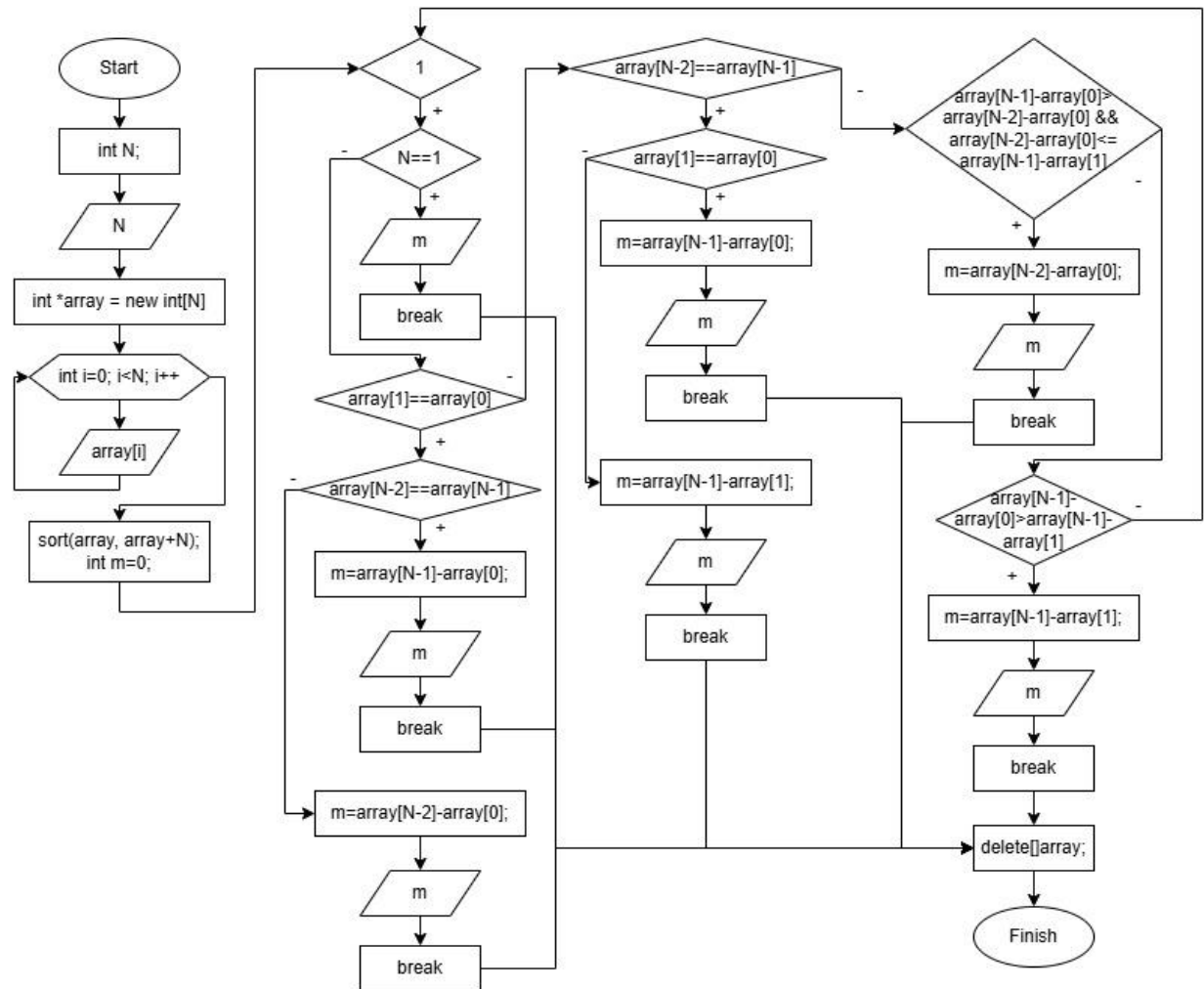
У першому рядку ціле число N - кількість чисел

У другому рядку масив r , який складається з N цілих чисел

Вихідні дані

Єдине ціле число m - мінімальна втома, яку можна отримати

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/41645432c95a57c2a05b55a5787ab90ee58838c1/ai_12/oleksandra_khvostova/epic_7/algotester_lab2_v1.cpp



```

1  #include <iostream>
2  #include <algorithm>
3  using namespace std;
4
5  int main(){
6      int N;
7      cin>>N;
8      int *array = new int[N];
9      for (int i=0; i<N; i++){
10         cin>>array[i];
11     }
12     sort(array, array+N);
13     int m=0;
14     while(1){
15         if(N==1) {cout<<m; break;}
16         if(array[1]==array[0]) {
17             if(array[N-2]==array[N-1]){
18                 m=array[N-1]-array[0];
19                 cout<<m;
20                 break;
21             }
22             m=array[N-2]-array[0];
23             cout<<m;
24             break;
25         }
26         if(array[N-2]==array[N-1]){
27             if(array[1]==array[0]){
28                 m=array[N-1]-array[0];
29                 cout<<m;
30                 break;
31             }
32             m=array[N-1]-array[1];
33             cout<<m;
34             break;}
35
36
37         if(array[N-1]-array[0]>array[N-2]-array[0]&&array[N-2]-array[0]<=array[N-1]-array[1]){
38             m=array[N-2]-array[0];
39             cout<<m;
40             break;
41         }
42         else if(array[N-1]-array[0]>array[N-1]-array[1]){
43             m=array[N-1]-array[1];
44             cout<<m;
45             break;
46         }}
47
48     delete[]array;
49     return 0;
50 }
51

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)
9 днів тому	C++ 23	Зараховано	0.003	1.285

Очікуваний час виконання: 1 година

Реальний час виконання: 2 години

Завдання №7. Algotester. Lab4v1

Lab 4v1

Обмеження: 1 сек., 256 MiB

Вам дано 2 цілих чисел масиви, розміром N та M .

Ваше завдання вивести:

1. Різницю $N-M$
2. Різницю $M-N$
3. Їх перетин
4. Їх об'єднання
5. Їх симетричну різницю

Вхідні дані

У першому рядку ціле число N - розмір масиву 1

У другому рядку N цілих чисел - елементи масиву 1

У третьому рядку ціле число M - розмір масиву 2

У четвертому рядку M цілих чисел - елементи масиву 2

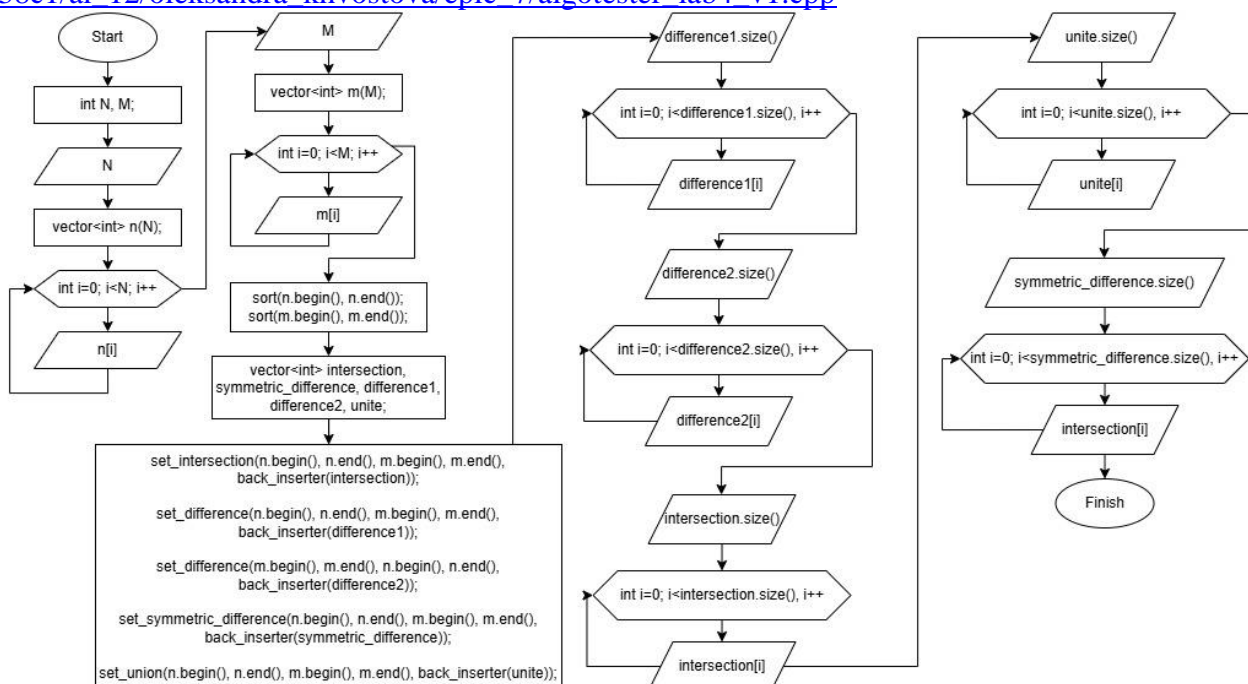
Вихідні дані

Вивести результат виконання 5 вищезазначених операцій у форматі:

У першому рядку ціле число N - розмір множини

У наступному рядку N цілих чисел - посортована у порядку зростання множина

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/41645432c95a57c2a05b55a5787ab90ee58838c1/ai_12/oleksandra_khvostova/epic_7/algotester_lab4_v1.cpp



```

1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 #include <iterator>
5
6 using namespace std;
7
8 int main() {
9     int N, M;
10    cin >> N;
11    vector<int> n(N);
12    for (int i = 0; i < N; i++) {
13        cin >> n[i];
14    }
15
16    cin >> M;
17    vector<int> m(M);
18    for (int i = 0; i < M; i++) {
19        cin >> m[i];
20    }
21    sort(n.begin(), n.end());
22    sort(m.begin(), m.end());
23
24    vector<int> intersection, symmetric_difference, difference1, difference2, unite;
25
26    set_intersection(n.begin(), n.end(), m.begin(), m.end(), back_inserter(intersection));
27    set_difference(n.begin(), n.end(), m.begin(), m.end(), back_inserter(difference1));
28    set_difference(m.begin(), m.end(), n.begin(), n.end(), back_inserter(difference2));
29    set_symmetric_difference(n.begin(), n.end(), m.begin(), m.end(), back_inserter(symmetric_difference));
30    set_union(n.begin(), n.end(), m.begin(), m.end(), back_inserter(unite));
31
32    cout << difference1.size() << endl;
33    for (int i=0; i<difference1.size(); i++) {
34        cout << difference1[i] << " ";
35    }
36    cout << endl<<endl;
37
38    cout << difference2.size() << endl;
39    for (int i=0; i<difference2.size(); i++) {
40        cout << difference2[i] << " ";
41    }
42    cout << endl<<endl;
43
44    cout << intersection.size() << endl;
45    for (int i=0; i<intersection.size(); i++) {
46        cout << intersection[i] << " ";
47    }
48    cout << endl<<endl;
49
50    cout << unite.size() << endl;
51    for (int i=0; i<unite.size(); i++) {
52        cout << unite[i] << " ";
53    }
54    cout << endl<<endl;
55
56    cout << symmetric_difference.size() << endl;
57    for (int i=0; i<symmetric_difference.size(); i++) {
58        cout << symmetric_difference[i] << " ";
59    }
60    cout << endl<<endl;
61
62    return 0;
63 }
64

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)
2 дні тому	C++ 23	Зараховано	0.003	1.230

Очікуваний час виконання: 1 година

Реальний час виконання: 1 година

Завдання №8. Algotester. Lab6v1

Lab 6v1

Обмеження: 2 сек., 256 MiB

Вам дано N слів та число K .

Ваше завдання перерахувати букви в словах, які зустрічаються в тексті більше-рівне ніж K разів (саме слово, не буква!).

Великі та маленькі букви вважаються однаковими, виводити необхідно малі, посортовані від останньої до першої у алфавіті. Букву потрібно виводити лише один раз.

У випадку якщо таких букв немає - вивести "Empty!".

Вхідні дані

Цілі числа N та K - загальна кількість слів та мінімальна кількість слів щоб враховувати букви цього слова в результаті.

N стрічок s

Вихідні дані

У першому рядку ціле число M - кількість унікальних букв

У другому рядку унікальні букви через пробіли

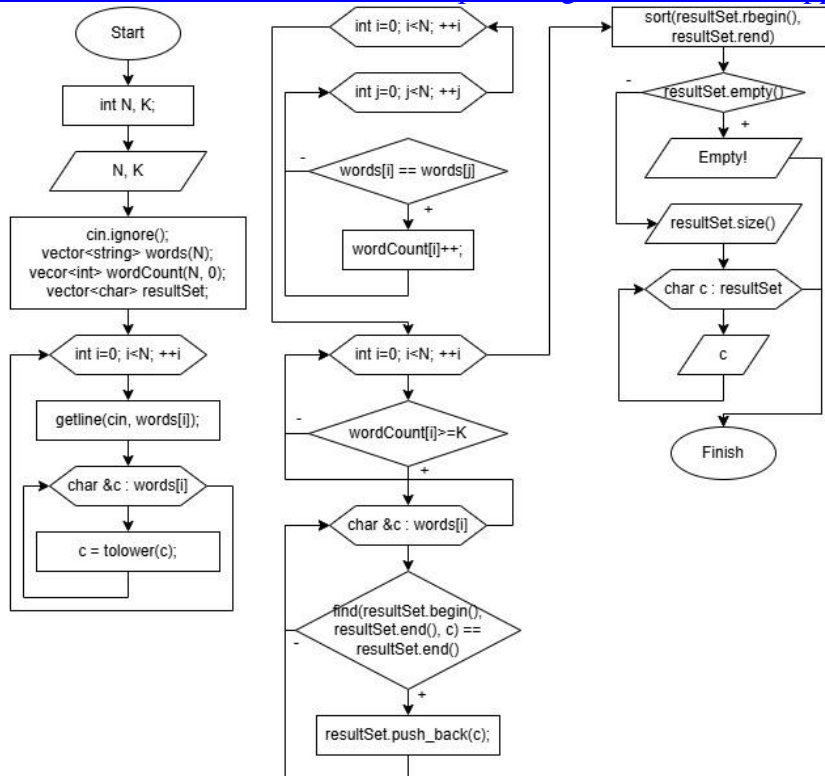
Обмеження

$$1 \leq K \leq N \leq 10^5$$

$$1 \leq |s_i| \leq 10$$

$$s_i \in a..z$$

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/41645432c95a57c2a05b55a5787ab90ee58838c1/ai_12/oleksandra_khvostova/epic_7/algotester_lab6_v1.cpp



```

1  #include <iostream>
2  #include <vector>
3  #include <unordered_map>
4  #include <unordered_set>
5  #include <algorithm>
6  #include <cctype>
7
8  using namespace std;
9
10 int main() {
11     int N, K;
12     cin >> N >> K;
13     cin.ignore(); // Ignore the newline after reading integers
14
15     unordered_map<string, int> wordCount;
16     unordered_set<char> resultSet;
17
18     // Read words and count their occurrences
19     for (int i = 0; i < N; ++i) {
20         string word;
21         getline(cin, word);
22
23         // Convert to lowercase and count the word
24         for (char &c : word) c = tolower(c);
25         wordCount[word]++;
26     }
27
28     // Check words meeting the threshold K and add their letters to the result set
29     for (const auto &pair : wordCount) {
30         if (pair.second >= K) {
31             for (char c : pair.first) resultSet.insert(c);
32         }
33     }
34
35     // Prepare and sort the result
36     vector<char> result(resultSet.begin(), resultSet.end());
37     sort(result.rbegin(), result.rend()); // Sort in reverse order
38
39     // Output results
40     if (result.empty()) {
41         cout << "Empty!" << endl;
42     } else {
43         cout << result.size() << endl;
44         for (char c : result) cout << c << " ";
45         cout << endl;
46     }
47
48     return 0;
49 }
50

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)
8 днів тому	C++ 23	Зараховано	0.035	1.313

Очікуваний час виконання: 1 година

Реальний час виконання: 1.5 години

Висновки: У ході виконання розрахункової роботи здобула практичні навички в розробці лінійних алгоритмів, застосуванні циклів, написанні власних функцій.