Министерство цифрового развития, связи и массовых коммуникаций

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»



Отчет по задачам

по дисциплине «Структура и алгоритмы обработки данных»

Выполнил: студент группы

БВТ1902

Бубенцов Н.В.

Москва

2021 г

Оглавление

[Цель работы 3](#__RefHeading___Toc7147_3533111288)

[Задание 3](#__RefHeading___Toc7149_3533111288)

[Код программы 5](#__RefHeading___Toc7171_3533111288)

[Снимки экрана работы программ](#__RefHeading___Toc7173_3533111288) 17

[Вывод](#__RefHeading___Toc7175_3533111288) 19

## Цель работы

Задача 1. «Треугольник с максимальным периметром» Массив A состоит из целых положительных чисел ­ длин отрезков. Составьте из трех отрезков такой треугольник, чтобы его периметр был максимально возможным. Если невозможно составить треугольник с положительной площадью ­ функция возвращает 0.

Ограничения:

3 ⩽ len(A) ⩽ 10000

1 ⩽ A[i] ⩽ 106

Задача 2. «Максимальное число» Дан массив неотрицательных целых чисел nums. Расположите их в таком порядке, чтобы вместе они образовали максимально возможное число. Замечание: Результат может быть очень большим числом, поэтому представьте его как string, а не integer.

Ограничения:

1 ⩽ len(nums) ⩽ 100

0 ⩽ nums[i] ⩽ 109 1

Задача 3. «Сортировка диагоналей в матрице» Дана матрица mat размером m \* n, значения ­ целочисленные. Напишите функцию, сортирующую каждую диагональ матрицы по возрастанию и возвращающую получившуюся матрицу.

Ограничения:

m == len(mat) • n == len(mat[i])

1 ⩽ m, n ⩽ 100

1 ⩽ mat[i][j] ⩽ 100

ЗАДАЧА 4 Даны две строки: s1 и s2 с одинаковым размером, проверьте, может ли некоторая перестановка строки s1 “победить” некоторую перестановку строки s2 или наоборот. Строка x может “победить” строку y (обе имеют размер n), если x[i]> = y [i] (в алфавитном порядке) для всех i от 0 до n-1. Примеры: Input: s1 = "abc", s2 = "xya" Output: true Объяснение: «ayx» - это перестановка строки s2 = «xya», которая “побеждает” строку s1 = «abc». Input: s1 = "abe", s2 = "acd" Output: false Объяснение: Все перестановки для s1 = "abe": "abe", "aeb", "bae", "bea", "eab" и "eba", а все перестановки для s2 = "acd": "acd", «adc», «cad», «cda», «dac» и «ca». Однако нет никакой перестановки строки s1, которая может нарушить некоторую перестановку строки s2 и наоборот. s1.length == n s2.length == n 1 <= n <= 10^5

ЗАДАЧА 5 Дана строка s, вернуть самую длинную полиндромную подстроку в s.

ЗАДАЧА 6 Вернуть количество отдельных непустых подстрок текста, которые могут быть записаны как конкатенация некоторой строки с самой собой (т.е. она может быть записана, как a + a, где a - некоторая строка).

Задача 7. «Стопки монет» На столе стоят 3n стопок монет. Вы и ваши друзья Алиса и Боб забираете стопки монет по следующему алгоритму: 1. Вы выбираете 3 стопки монет из оставшихся на столе. 2. Алиса забирает себе стопку с максимальным количеством монет. 3. Вы забираете одну из двух оставшихся стопок. 4. Боб забирает последнюю стопку. 5. Если еще остались стопки, то действия повторяются с первого шага. Дан массив целых положительных чисел piles. Напишите функцию, возвращающую максимальное число монет, которое вы можете получить.

Ограничения:

3 ⩽ len(piles) ⩽ 105

len(piles) mod 3 == 0

1 ⩽ piles[i] ⩽ 104

Задача 8. «Шарики и стрелы» Некоторые сферические шарики распределены по двухмерному пространству. Для каждого шарика даны x­координаты начала и конца его горизонтального диаметра. Так как пространство двумерно, то y­координаты не имеют значения в данной задаче. Координата xstart всегда меньше xend. Стрелу можно выстрелить строго вертикально (вдоль y­оси) из разных точек x­оси. Шарик с координатами xstart и xend уничтожается стрелой, если она была выпущена из такой позиции x, что xstart ⩽ x ⩽ xend. Когда стрела выпущена, она летит в пространстве бесконечное время (уничтожая все шарики на пути). Дан массив points, где points[i] = [xstart, xend]. Напишите функцию, возвращающую минимальное количество стрел, которые нужно выпустить, чтобы уничтожить все шарики.

Ограничения:

0 ⩽ len(points) ⩽ 104

len(points[i]) == 2

−2 31 ⩽ xstart < xend ⩽ 2 31 − 1

## Код программы

Задача 1 — 3   
#include <iostream>  
#include <vector>  
#include <algorithm>  
  
using namespace std;  
  
std::vector<int> generate(int rows, int min, int max)  
{  
 std::vector<int> ans;  
 int el;  
 for (int i = 0; i < rows; ++i)  
 {  
 el=min+ rand()%max;  
 ans.push\_back(el);  
 }  
 return ans;  
}  
  
int\*\* generate(int rows, int cols, int min, int max)  
{  
 int\*\* matrix= new int\*[rows];  
 for(int i=0; i<rows; ++i)  
 matrix[i]=new int[cols];  
  
 for(int i=0; i<rows; ++i)  
 {  
 for (int j = 0; j < cols; ++j)  
 {  
 matrix[i][j]=min+rand()% max;  
 }  
 }  
 return matrix;  
}  
  
  
int triangle(vector<int> arr)  
{  
 if (arr.size()<3)  
 return 0;  
  
 vector<int>::iterator iterator= arr.end();  
 int sides[3];  
  
  
 for (int & side : sides)  
 {  
 iterator--;  
 side=\*iterator;  
 }  
  
 while (sides[0]>=sides[1]+sides[2])  
 {  
 if (iterator == arr.begin())  
 {  
 return 0;  
 }  
 else  
 {  
 iterator--;  
 sides[0] = sides[1];  
 sides[1] = sides[2];  
 sides[2] = \*iterator;  
 }  
 }  
 return (sides[0]+sides[1]+sides[2]);  
}  
  
  
std::string maxNumber (std::vector<int> arr) // 3 39 9 9393  
{  
 std::vector<std::string> strings;  
  
 for (auto num:arr)  
 {  
 strings.emplace\_back(std::to\_string(num));  
 }  
  
 std::sort(  
 strings.begin(),  
 strings.end(),  
 [](auto const &L, auto const &R)  
 {  
 return R + L < L + R ;  
 }  
 );  
 std::string res\_str;  
 for (auto const &str: strings)  
 {  
 res\_str += str;  
 }  
  
 return res\_str;  
}  
  
  
int\*\* sort\_matrix(int\*\* matrix, int n, int m)  
{  
 int j=0;  
 int i;  
 std::vector<int> arr;  
 std::vector<int>::iterator iterator;  
  
 for (int rows=0; rows<m; rows++)  
 {  
 i=rows;  
 arr.erase(arr.begin(), arr.end());  
  
 while (i-j<n && i<m)  
 {  
 arr.push\_back(matrix[i-j][i]);  
 i++;  
 }  
  
 sort(arr.begin(),arr.end());  
  
 i=rows;  
 iterator =arr.begin();  
  
 while (i-j<n && i<m)  
 {  
 matrix[i-j][i]=\*iterator;  
 i++;  
 iterator++;  
 }  
 j++;  
 }  
  
 i=0;  
  
 for (int line=0; line<n; line++)  
 {  
 j=line;  
 arr.erase(arr.begin(), arr.end());  
  
 while (j<n && j-i<m)  
 {  
 arr.push\_back(matrix[j][j-i]);  
 j++;  
 }  
  
 sort(arr.begin(),arr.end());  
  
 iterator =arr.begin();  
 j=line;  
  
 while (j<n && j-i<m)  
 {  
 matrix[j][j-i]=\*iterator;  
 j++;  
 iterator++;  
 }  
 i++;  
 }  
  
  
 return matrix;  
}  
  
int main()  
{  
 cout << "Первая задача" << endl;  
 int n;  
 int m;  
 int len;  
 int temp;  
 int min;  
 int max;  
 vector<int> arr;  
  
 cout << "Введите кол-во элементов" << endl;  
 cin >> len;  
 cout << "Введите элементы массива" << endl;  
 for (int i = 0; i < len; ++i)  
 {  
 cin >> temp;  
 arr.push\_back(temp);  
 }  
  
 sort(arr.begin(), arr.end());  
 cout << triangle(arr) << endl;  
  
  
 cout << "Вторая задача" << endl;  
  
 arr.erase(arr.begin(), arr.end());  
  
 cout << "Введите кол-во элементов" << endl;  
 cin >> len;  
 cout << "Введите минимальное возможное положительное значение" << endl;  
 cin>> min;  
 cout << "Введите максимальное возможное положительное значение" << endl;  
 cin>>max;  
 arr=generate(len, min, max);  
  
 cout << "Сгенерированный массив:";  
 for (auto i: arr)  
 cout << i<< " ";  
 cout<< endl;  
  
 std::cout<< maxNumber(arr)<< endl;  
  
  
 cout << "Третья задача" << endl;  
 cout << "Введите количество строк" << endl;  
 cin >> n;  
 cout << "Введите количество столбцов" << endl;  
 cin >> m;  
 cout << "Введите минимальное возможное значение" << endl;  
 cin>> min;  
 cout << "Введите максимальное возможное значение" << endl;  
 cin>>max;  
  
 int \*\*matrix = new int \*[n];  
 for (int i = 0; i < n; i++)  
 matrix[i] = new int[m];  
  
 matrix=generate(n, m, min, max);  
  
 cout<< "Сгенерированная матрица: "<<endl;  
  
 for (int i = 0; i < n; ++i)  
 {  
 for (int j = 0; j < m; ++j)  
 {  
 cout << matrix[i][j] << " ";  
 }  
 cout << endl;  
 }  
  
 matrix = sort\_matrix(matrix, n, m);  
  
 cout<< "Отсортированная матрица: "<<endl;  
  
  
 for (int i = 0; i < n; ++i)  
 {  
 for (int j = 0; j < m; ++j)  
 {  
 cout << matrix[i][j] << " ";  
 }  
 cout << endl;  
 }  
}

Задача 4 — 6

#include "tasks.cpp"  
  
using namespace std;  
  
  
int main()  
{  
 task1();  
  
 task2();  
  
 task3();  
  
 return 0;  
}

#include <iostream>  
#include <string>  
#include <algorithm>  
  
using namespace std;  
  
void task1()  
{  
 string s1;  
 string s2;  
 bool aa=true;  
 bool bb=true;  
  
 cout <<"Введите первую строку"<< endl;  
 cin >> s1;  
 cout <<"Введите вторую строку"<< endl;  
 cin >> s2;  
  
 sort(s1.begin(), s1.end());  
 sort(s2.begin(), s2.end());  
  
 for (int i=0; i<s1.size() && (aa || bb); i++)  
 {  
 if (s1[i] < s2[i])  
 aa = false;  
 else if (s2[i] < s1[i])  
 bb = false;  
 }  
 cout << (aa|| bb) << endl;  
}  
  
void task2()  
{  
 string a;  
 string buffer;  
 string sub\_a;  
 string ans;  
  
 cout << "Введите строку" << endl;  
 cin >> a;  
  
  
 for (char i : a)  
 {  
 sub\_a += i;  
 buffer = sub\_a;  
 reverse(buffer.begin(), buffer.end());  
 if (sub\_a == buffer)  
 {  
 ans = sub\_a;  
 }  
 }  
 cout << ans << endl;  
}  
  
void task3()  
{  
 bool first=true;  
 int ans=0;  
 string a;  
 string substring;  
 string buffer="";  
  
 cout << "Введите строку" << endl;  
 cin >> a;  
  
  
  
 for (int i=0; (a.size() / 2) > i; i++)  
 {  
 buffer="";  
  
 if (first)  
 {  
 for (int j = 0; j <= (a.size() / 2); j++)  
 {  
 buffer += a[j];  
 substring = buffer + buffer;  
 if (a.find(substring) != -1)  
 ans++;  
 }  
 first=false;  
 }  
 buffer="";  
  
 for (int j = i+1; j <= (a.size() / 2); j++)  
 {  
 buffer += a[j];  
 substring = buffer + buffer;  
 if (a.find(substring) != -1)  
 ans++;  
 }  
 }  
 cout<< ans;  
}  
Задача 7

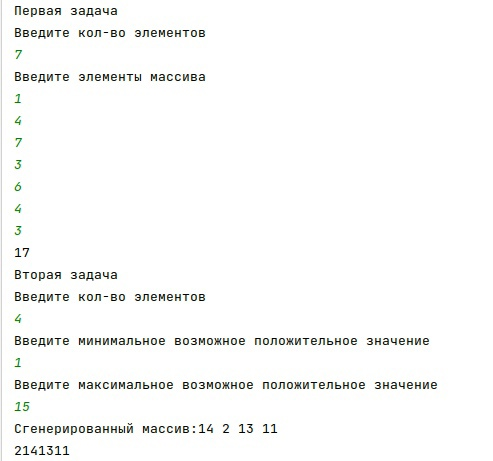
#include <iostream>  
#include <vector>  
#include <algorithm>  
  
using namespace std;  
  
int main()  
{  
 vector<int> coins;  
 int len;  
 int a;  
 int ans=0;  
  
 cout<< "Введите кол-во монет"<<endl;  
 cin >>len;  
  
 cout <<"Введите монеты через enter"<<endl;  
  
 for (int i=0; i<len; ++i)  
 {  
 cin>>a;  
 coins.push\_back(a);  
 }  
  
 sort(coins.begin(), coins.end());  
  
 while (!coins.empty())  
 {  
 coins.erase(coins.end()-1);  
 ans+= \*(coins.end()-1);  
 coins.erase(coins.end()-1);  
 coins.erase(coins.begin());  
 }  
 cout<<ans;  
 return 0;  
}

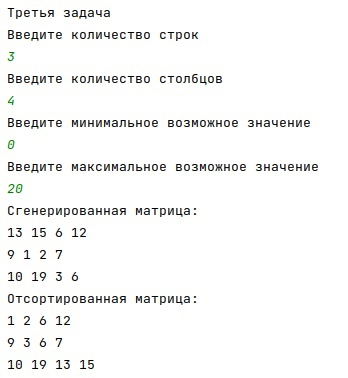
Задача 8

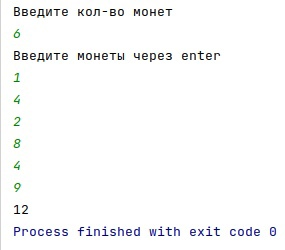
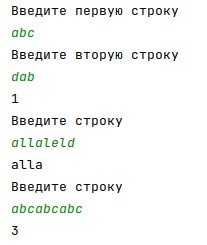
#include <iostream>  
#include <vector>  
#include <algorithm>  
#include <cstdlib>  
#include "Ballon.h"  
  
using namespace std;  
  
int main()  
{  
 int amount;  
 int arrow;  
 int count = 0;  
 int temp = 0;  
 vector<ballon> ballons;  
  
 srand(time(nullptr));  
  
 cout << "Введите кол-во шаров" << endl;  
 cin >> amount;  
  
 for (int i = 0; i < amount; i++)  
 {  
 ballons.push\_back(ballon::generator());  
 }  
 cout << "Сгенерированные шары: ";  
  
 for (auto i: ballons)  
 cout << i.getStart() << " " << i.getEnd() << "; ";  
 cout << endl;  
  
 ballon::sort(ballons);  
  
  
  
  
 while (temp != ballons.size() - 1)  
 {  
 arrow = ballons[temp].getEnd();  
 while (arrow >= ballons[temp].getStart() && arrow <= ballons[temp].getEnd())  
 {  
 if (temp == ballons.size() - 1)  
 break;  
  
 count++;  
 temp++;  
 }  
 }  
 cout << "Минимальное количество стрел: " << count;  
  
 return 0;  
}

#include <cstdlib>  
#include <vector>  
  
#ifndef BALLON\_BALLON\_H  
#define BALLON\_BALLON\_H  
  
  
class ballon  
{  
private:  
 int startx;  
 int endx;  
  
public:  
  
 ballon(){};  
  
 ballon(int start, int end)  
 {  
 startx=start;  
 endx=end;  
 }  
  
 int getStart() { return startx; }  
  
 int getEnd() { return endx; }  
  
 void setStart(int x) { startx=x; }  
  
 void setEnd(int x) { endx=x; }  
  
 static ballon generator()  
 {  
 int start=rand()% 20;  
 int end=start + rand()% 5;  
 return ballon(start, end);  
 }  
  
 static void sort (std::vector<ballon>& ballons)  
 {  
 for (int i=0; i<ballons.size()-1; i++)  
 {  
 for (int j=i+1; j<ballons.size(); j++)  
 {  
 if (ballons[i].getEnd() > ballons[j].getEnd())  
 {  
 std::swap(ballons[i], ballons[j]);  
 }  
 }  
 }  
 }  
};

## Снимки экрана работы программ







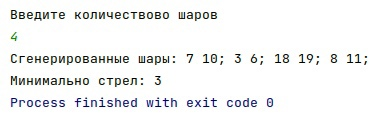


Рисунок 1 – Результаты работы программы

## Вывод

Я реализовал в программном коде на языке С++ решения 8 задач.