**CÁCH ĐÁNH GIÁ ĐIỂM THỰC HÀNH**

**HỌC PHẦN: IT3040 - KỸ THUẬT LẬP TRÌNH – 2022.2**

1. **Quy định, yêu cầu:** 
   * Tài liệu và nội dung thực hành chấm điểm trên hệ thống:

https://lab.soict.hust.edu.vn/

* + Bài tập trên lớp chấm điểm tự động (các bài không chấm trên hệ thống làm vào máy tính 🡺 làm báo cáo thực hành – Theo mẫu).
  + Hạn nộp báo cáo trên Teams (Bài tập trên lớp + Bài tập về nhà): 1 tuần.

1. **Đánh giá điểm thực hành**

1. Chuyên cần (đúng giờ, nghiêm túc trong giờ học) - Điểm danh trên Teams: 10%

2. Báo cáo thực hành (bài tập trên lớp + Về nhà) theo mẫu nộp trên Teams: 40%

3. Trắc nghiệm – Form trên Teams: 10%

4. Kiểm tra thực hành: 40%. (Tiết 2,3 buổi thực hành thứ 5).

**Điểm thưởng: 5% 🡪 10% (Cho Mục 1,2 điểm TB từ 9-10).**

Tham gia thực hành đúng giờ đầy đủ theo thời khóa biểu (nếu có lý do không đi thực hành đúng kíp được thì gửi mail xin phép thực hành bù trước 1 ngày qua mail [hoalt@soict.hust.edu.vn](mailto:hoalt@soict.hust.edu.vn), Tiêu đề: đăng ký học bù – IT3040 – MaLopTH.

Các kíp có thể bù:

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Thời gian, địa điểm, Tuần học** | **Mã nhóm** | **Mã lớp** |
| **1** |  |  |  |
| **2** |  |  |  |
| **3** |  |  |  |
| **4** |  |  |  |
| **5** |  |  |  |
| **6** |  |  |  |
| **7** |  |  |  |

**Nếu nghỉ không có lý do 3 buổi, không thực hành bù thì điểm chuyên cần, báo cáo và BTVN coi như 0 điểm thực hành.**

**[Bài thực hành số 3 – Tuần 36 3](#_Toc13968)**

[Bài tập 3.1. Tính dãy Lucas 3](#_Toc13406)

[Bài tập 3.2. Quân mã đi tuần 5](#_Toc25999)

[Bài tập 3.3. Bài toán người du lịch 13](#_Toc2142)

[Bài tập 3.4. LIS 18](#_Toc20903)

[Bài tập 3.5. Tính tổ hợp 23](#_Toc30264)

[Bài tập 3.6. Tìm ước chung lớn nhất 30](#_Toc25807)

[Bài tập 3.7. Liệt kê xâu nhị phân 33](#_Toc17142)

[Bài tập 3.8. Cân đĩa 43](#_Toc16254)

[Bài tập 3.9. Lập lịch cho y tá 48](#_Toc20488)

[Bài tập 3.10. Khoảng cách Hamming 53](#_Toc2345)

[Bài tập 3.11. Lịch trình chụp ảnh 58](#_Toc1829)

[Bài tập 3.12. Đếm đường đi 67](#_Toc31507)

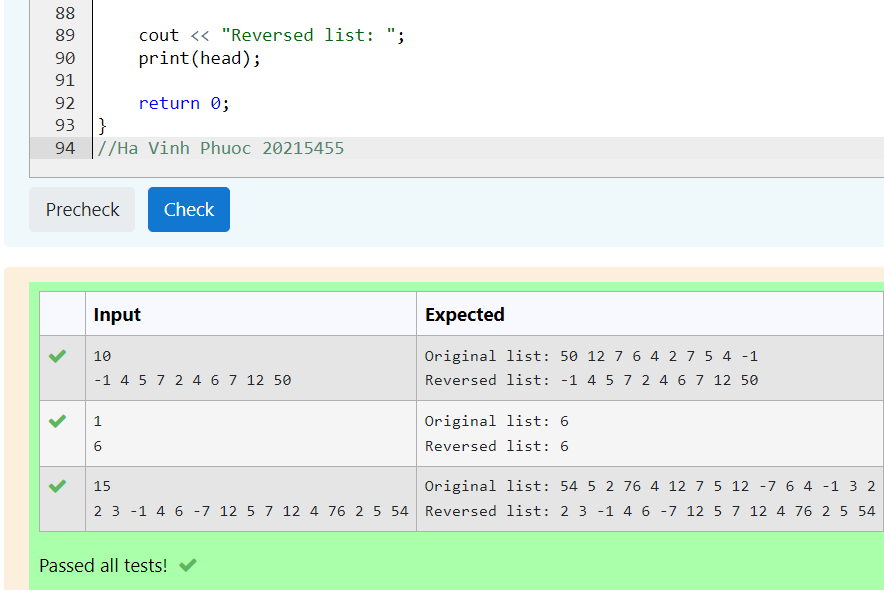
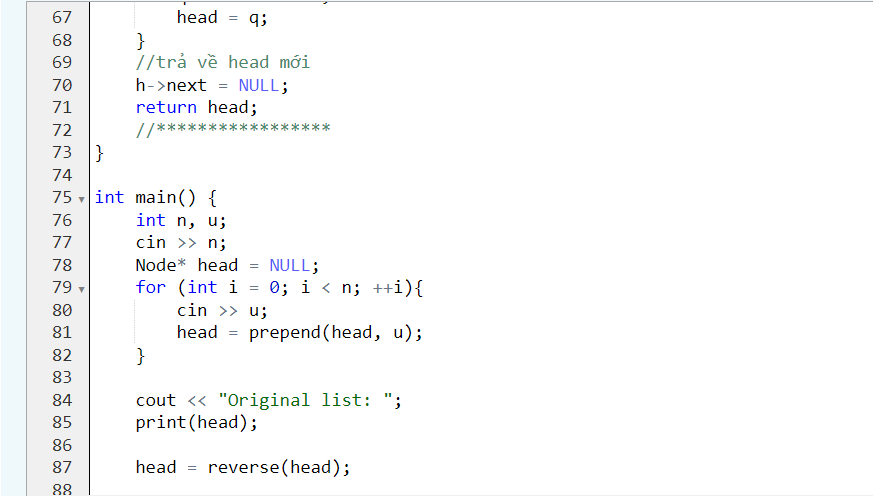
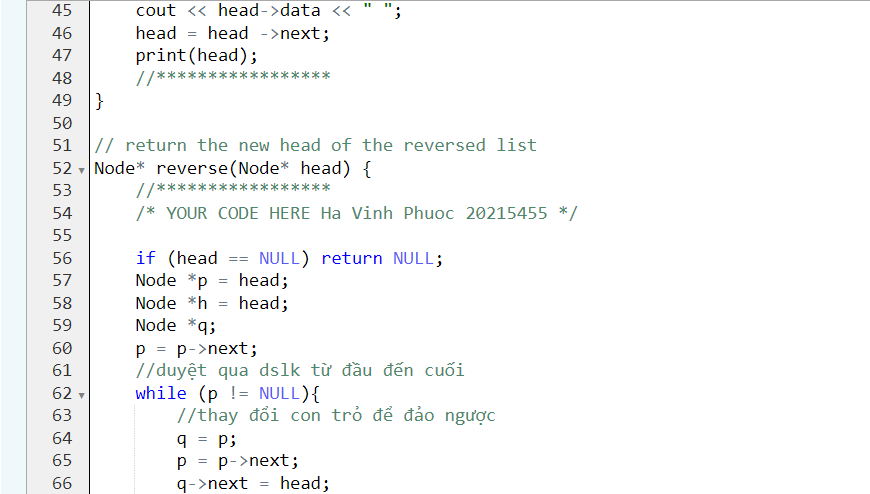
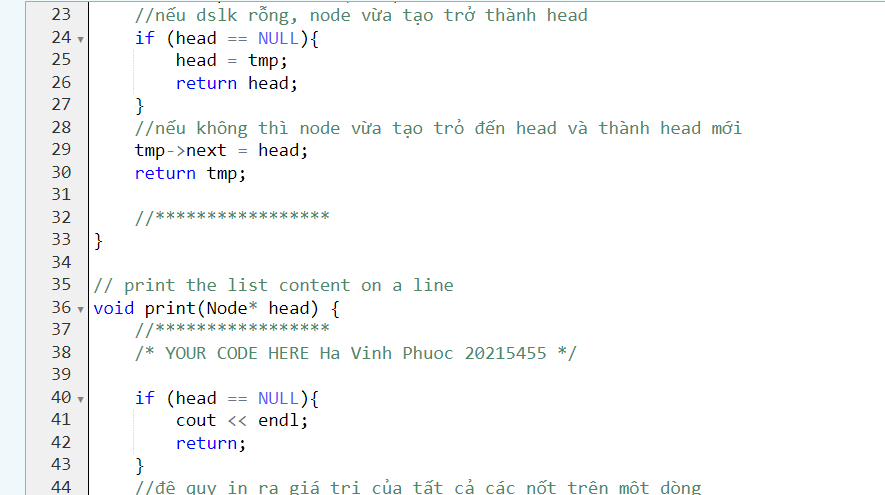
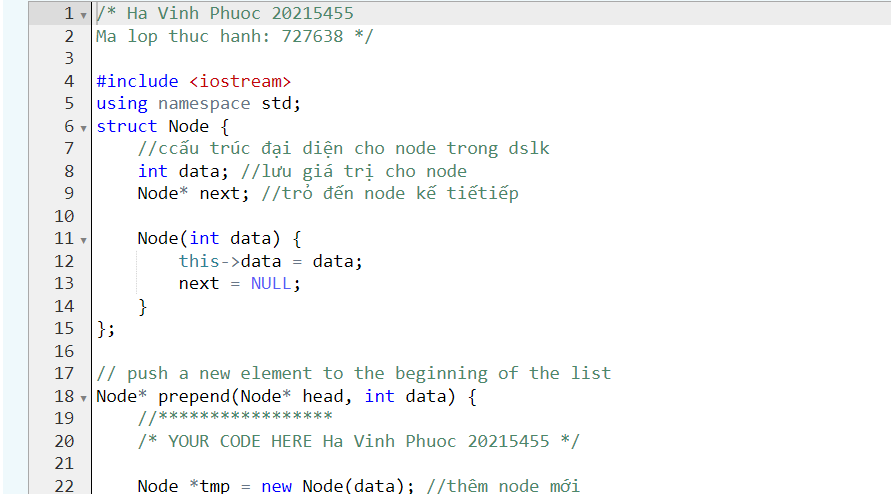
# Bài thực hành số 4 – Tuần 38

## **Bài tập 4.1. Đảo ngược một danh sách liên kết đơn**

## Hãy hoàn thiện các hàm thao tác trên một danh sách liên kết:

## Thêm một phần tử vào đầu danh sách liên kết In danh sách

## Đảo ngược danh sách liên kết (yêu cầu độ phức tạp thời gian O(N) và chi phí bộ nhớ dùng thêm O(1))



/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include <iostream>

using namespace std;

struct Node {

//ccấu trúc đại diện cho node trong dslk

int data; //lưu giá trị cho node

Node\* next; //trỏ đến node kế tiếtiếp

Node(int data) {

this->data = data;

next = NULL;

}

};

// push a new element to the beginning of the list

Node\* prepend(Node\* head, int data) {

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

Node \*tmp = new Node(data); //thêm node mới

//nếu dslk rỗng, node vừa tạo trở thành head

if (head == NULL){

head = tmp;

return head;

}

//nếu không thì node vừa tạo trỏ đến head và thành head mới

tmp->next = head;

return tmp;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

// print the list content on a line

void print(Node\* head) {

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

if (head == NULL){

cout << endl;

return;

}

//đệ quy in ra giá trị của tất cả các nốt trên một dòng

cout << head->data << " ";

head = head ->next;

print(head);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

// return the new head of the reversed list

Node\* reverse(Node\* head) {

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

if (head == NULL) return NULL;

Node \*p = head;

Node \*h = head;

Node \*q;

p = p->next;

//duyệt qua dslk từ đầu đến cuối

while (p != NULL){

//thay đổi con trỏ để đảo ngược

q = p;

p = p->next;

q->next = head;

head = q;

}

//trả về head mới

h->next = NULL;

return head;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

int main() {

int n, u;

cin >> n;

Node\* head = NULL;

for (int i = 0; i < n; ++i){

cin >> u;

head = prepend(head, u);

}

cout << "Original list: ";

print(head);

head = reverse(head);

cout << "Reversed list: ";

print(head);

return 0;

}

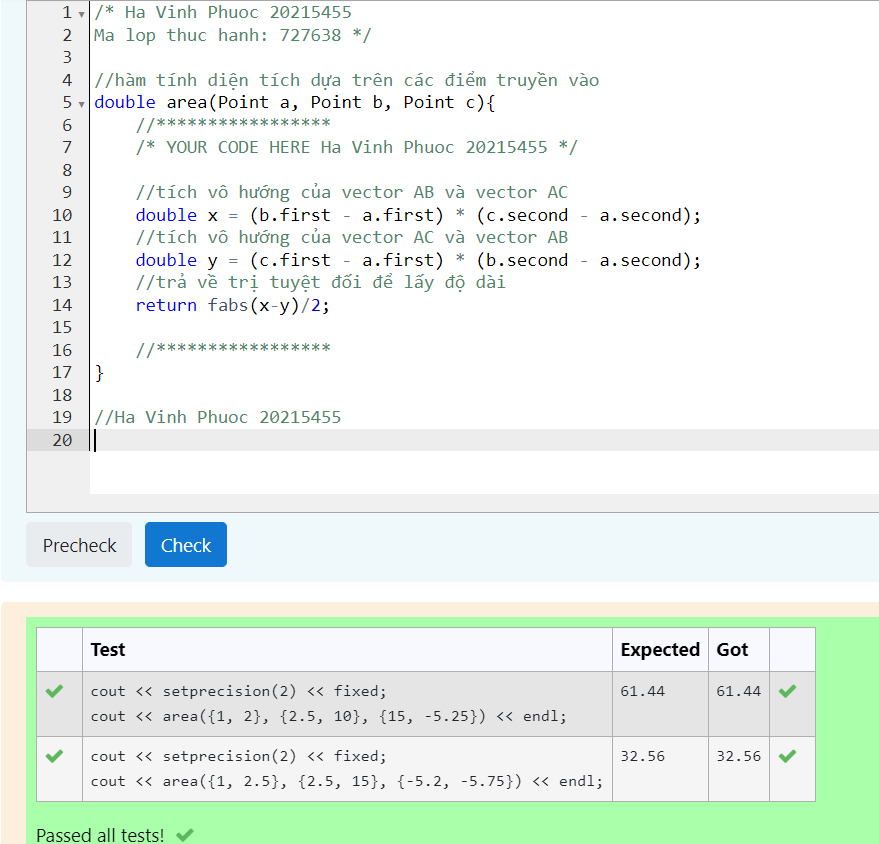
//Ha Vinh Phuoc 20215455

## **Bài tập 4.2. Tính diện tích tam giác**

## Một điểm trong không gian 2 chiều được biểu diễn bằng pair.

## Hãy viết hàm double area(Point a, Point b, Point c) tính diện tích tam giác theo tọa độ 3 đỉnh.

## Trong đó, Point là kiểu được định nghĩa sẵn trong trình chấm như sau: using Point = pair<double, double>;



/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include <iostream>

#include <cmath>

#include <iomanip>

#include <utility>

using namespace std;

using Point = pair<double, double>;

//hàm tính diện tích dựa trên các điểm truyền vào

double area(Point a, Point b, Point c){

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

//tích vô hướng của vector AB và vector AC

double x = (b.first - a.first) \* (c.second - a.second);

//tích vô hướng của vector AC và vector AB

double y = (c.first - a.first) \* (b.second - a.second);

//trả về trị tuyệt đối để lấy độ dài

return fabs(x-y)/2;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

int main() {

//in ra kết quả với 2 chữ số thập phân sau dấu phẩy

cout << setprecision(2) << fixed;

cout << area({1, 2}, {2.5, 10}, {15, -5.25}) << endl;

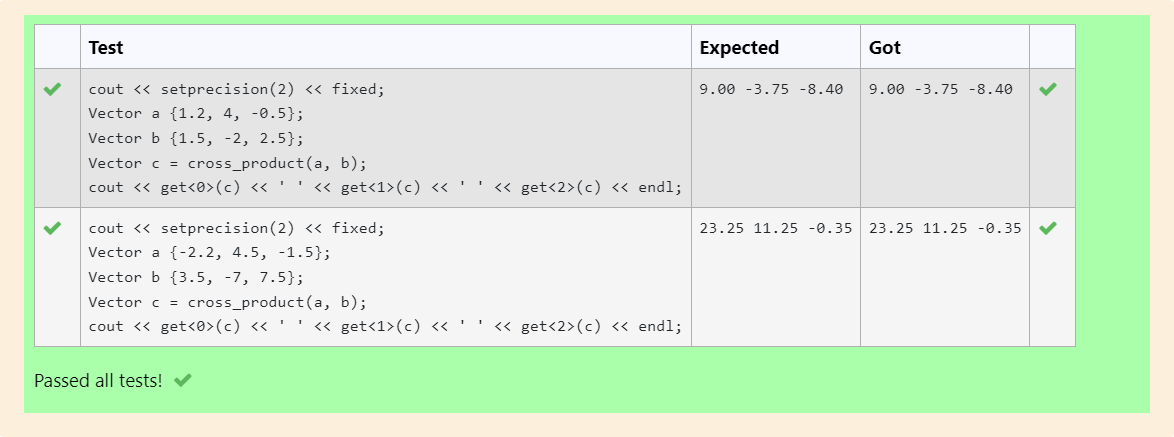
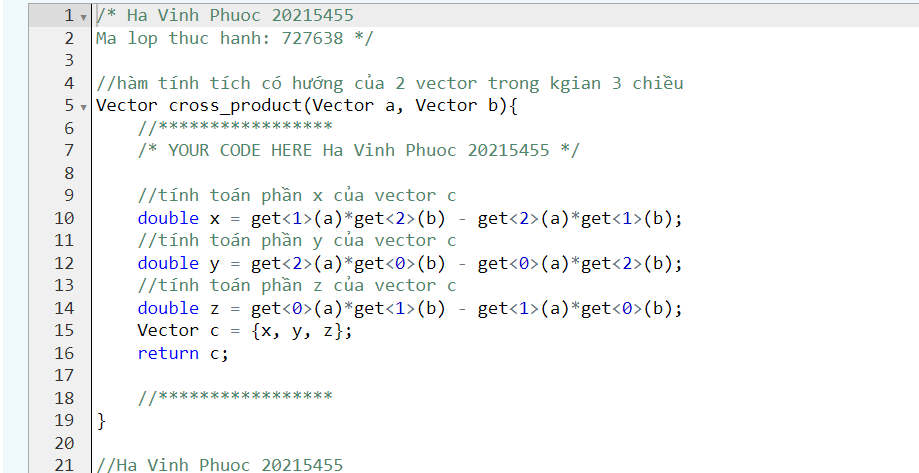
return 0;

}

//Ha Vinh Phuoc 20215455

## **Bài tập 4.3. Tính tích có hướng của 2 vector**

## Một vector trong không gian 3 chiều được biểu diễn bằng tuple<double, double, double>. Hãy viết hàm Vector cross\_product(Vector a, Vector b) tính tích có hướng của 2 vector. Trong đó Vector là kiểu dữ liệu được định nghĩa sẵn trong trình chấm như sau: using Vector = tuple<double, double, double>;



/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include<iostream>

#include<cmath>

#include<iomanip>

using namespace std;

using Vector = tuple<double, double, double>;

//hàm tính tích có hướng của 2 vector trong kgian 3 chiều

Vector cross\_product(Vector a, Vector b){

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

//tính toán phần x của vector c

double x = get<1>(a)\*get<2>(b) - get<2>(a)\*get<1>(b);

//tính toán phần y của vector c

double y = get<2>(a)\*get<0>(b) - get<0>(a)\*get<2>(b);

//tính toán phần z của vector c

double z = get<0>(a)\*get<1>(b) - get<1>(a)\*get<0>(b);

Vector c = {x, y, z};

return c;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

int main() {

cout << setprecision(2) << fixed;

Vector a {2.0, 4.0, -0.5};

Vector b {1.5, -2.0, 3.3};

Vector c = cross\_product(a, b);

cout << get<0>(c) << ' ' << get<1>(c) << ' ' << get<2>(c) << endl;

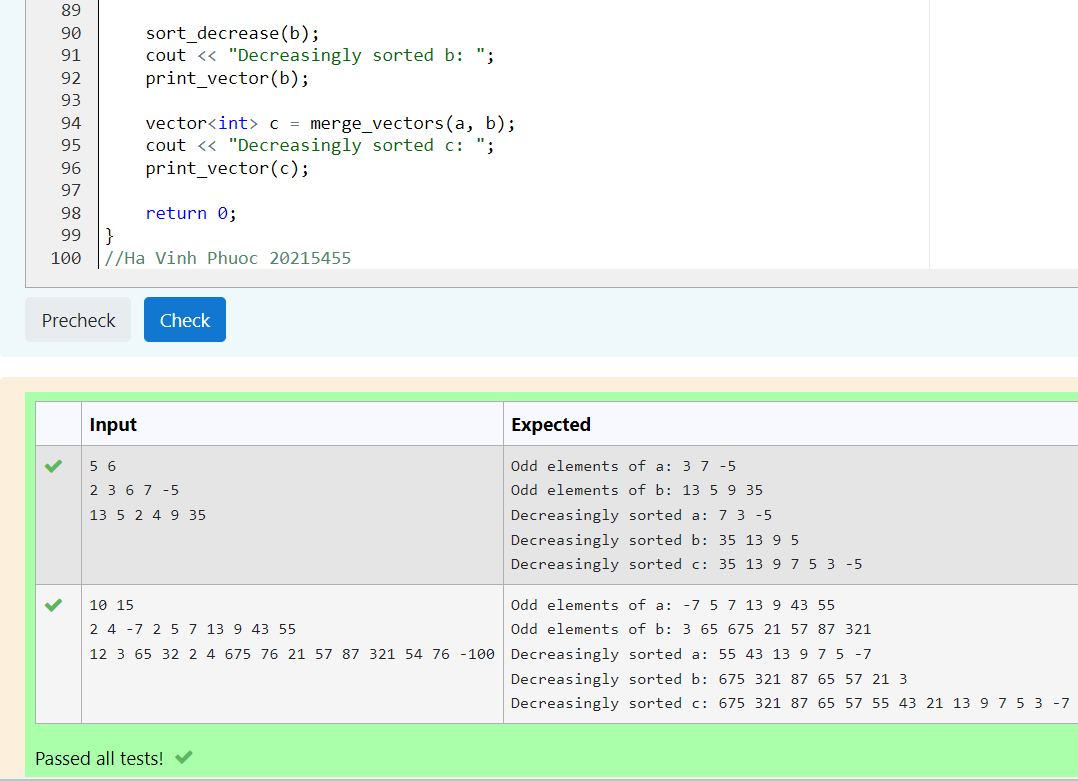
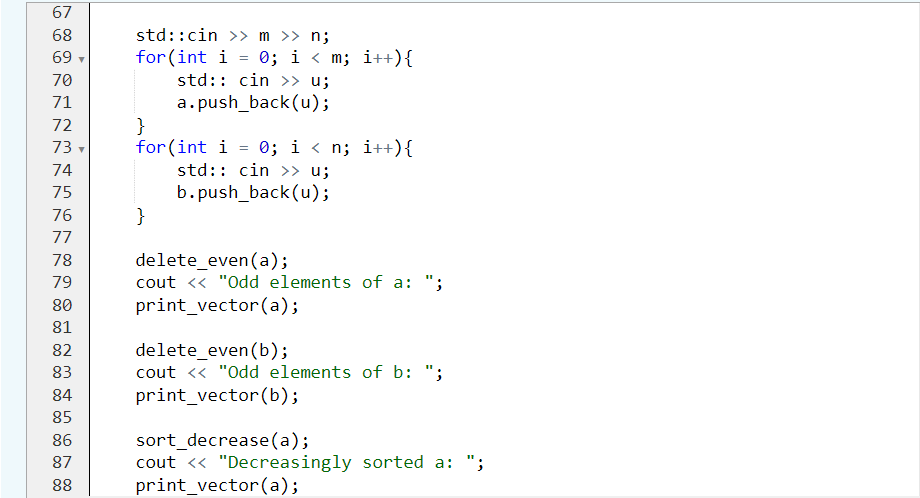
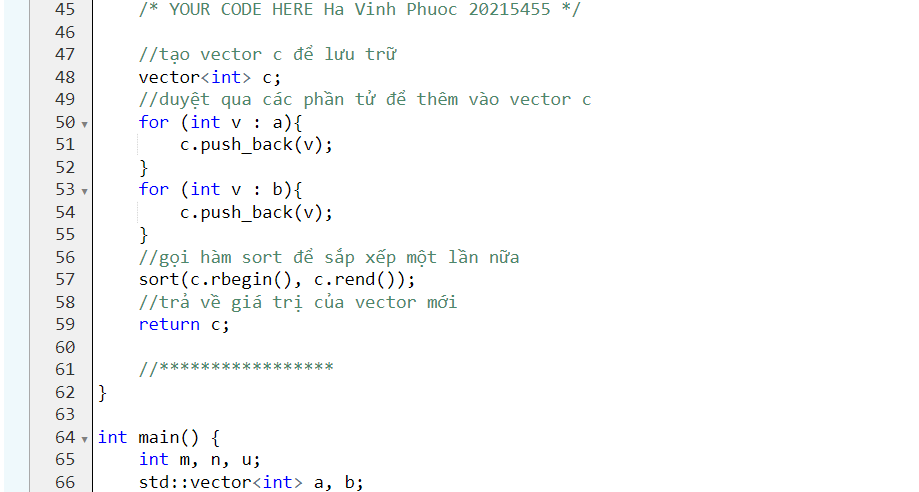
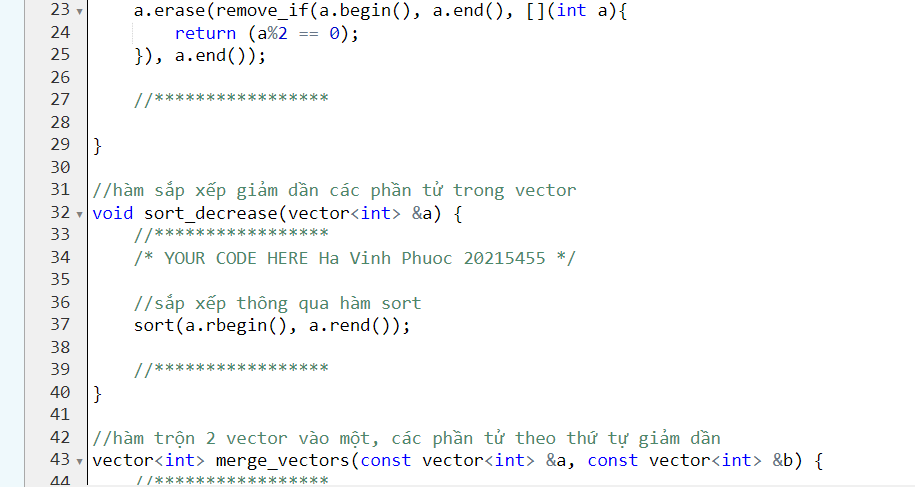
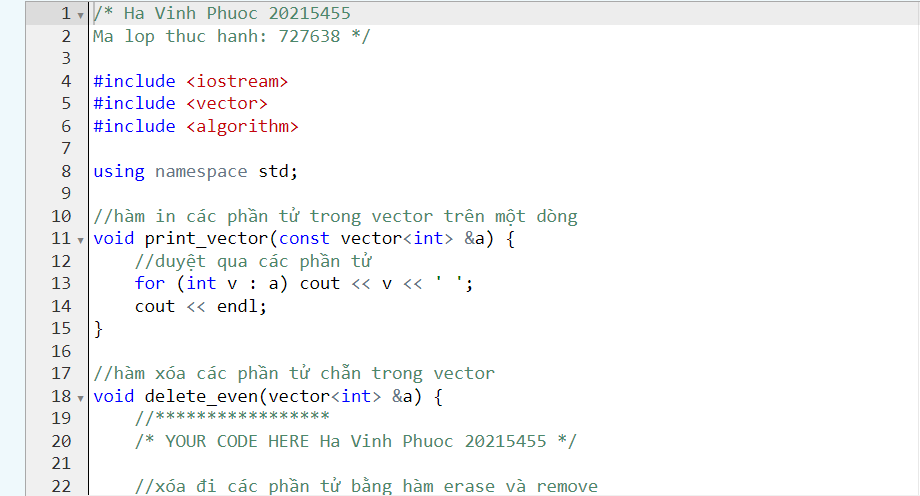
return 0;

}

//Ha Vinh Phuoc 20215455

## **Bài tập 4.4. Thao tác với vector**

## Cho hai vector, hãy xóa hết các phần tử chẵn, sắp xếp giảm dần các số trong cả 2 vector và trộn lại thành một vector cũng được sắp xếp giảm dần.



/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

//hàm in các phần tử trong vector trên một dòng

void print\_vector(const vector<int> &a) {

//duyệt qua các phần tử

for (int v : a) cout << v << ' ';

cout << endl;

}

//hàm xóa các phần tử chẵn trong vector

void delete\_even(vector<int> &a) {

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

//xóa đi các phần tử bằng hàm erase và remove

a.erase(remove\_if(a.begin(), a.end(), [](int a){

return (a%2 == 0);

}), a.end());

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

//hàm sắp xếp giảm dần các phần tử trong vector

void sort\_decrease(vector<int> &a) {

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

//sắp xếp thông qua hàm sort

sort(a.rbegin(), a.rend());

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

//hàm trộn 2 vector vào một, các phần tử theo thứ tự giảm dần

vector<int> merge\_vectors(const vector<int> &a, const vector<int> &b) {

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

//tạo vector c để lưu trữ

vector<int> c;

//duyệt qua các phần tử để thêm vào vector c

for (int v : a){

c.push\_back(v);

}

for (int v : b){

c.push\_back(v);

}

//gọi hàm sort để sắp xếp một lần nữa

sort(c.rbegin(), c.rend());

//trả về giá trị của vector mới

return c;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

int main() {

int m, n, u;

std::vector<int> a, b;

std::cin >> m >> n;

for(int i = 0; i < m; i++){

std:: cin >> u;

a.push\_back(u);

}

for(int i = 0; i < n; i++){

std:: cin >> u;

b.push\_back(u);

}

delete\_even(a);

cout << "Odd elements of a: ";

print\_vector(a);

delete\_even(b);

cout << "Odd elements of b: ";

print\_vector(b);

sort\_decrease(a);

cout << "Decreasingly sorted a: ";

print\_vector(a);

sort\_decrease(b);

cout << "Decreasingly sorted b: ";

print\_vector(b);

vector<int> c = merge\_vectors(a, b);

cout << "Decreasingly sorted c: ";

print\_vector(c);

return 0;

}

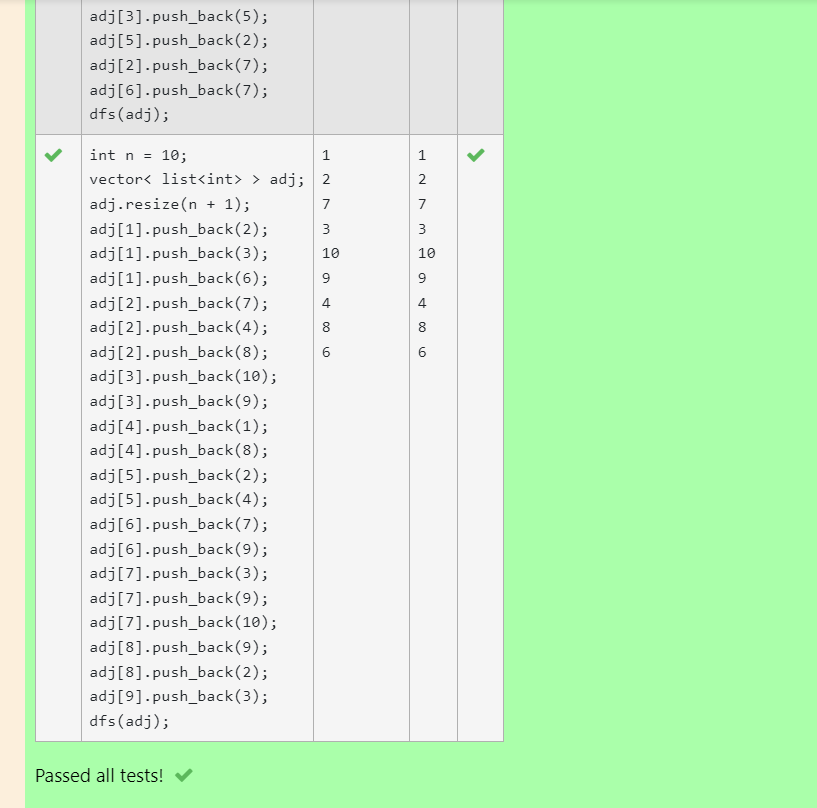
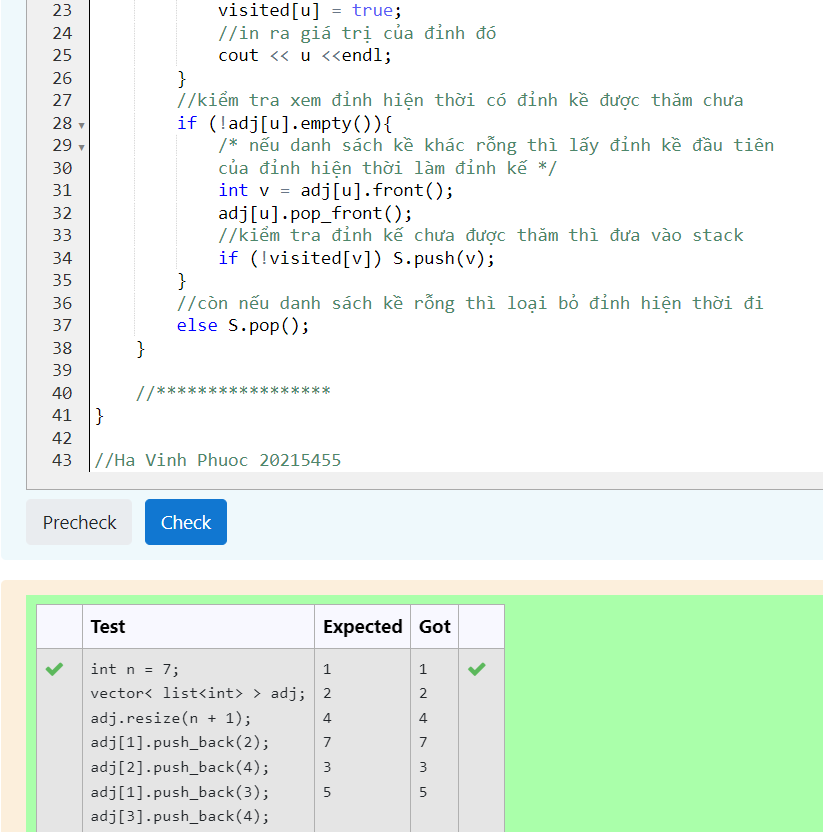
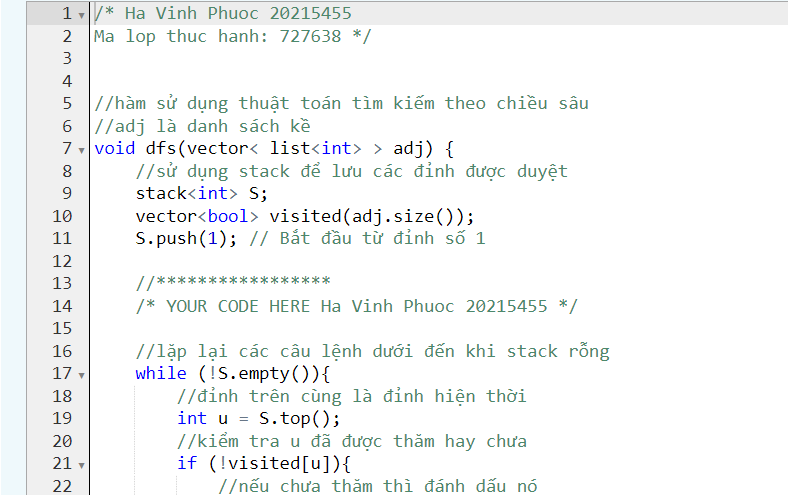
//Ha Vinh Phuoc 20215455

## **Bài tập 4.5.**

## Viết hàm thực hiện thuật toán DFS không sử dụng đệ quy trên đồ thị biểu diễn bằng danh sách kề vector< list<int> > .

## Đồ thị có n đỉnh được đánh số từ 1 đến n. Thuật toán DFS xuất phát từ đỉnh 1. Các đỉnh được thăm theo thứ tự ưu tiên từ trái sang phải trong danh sách kề.

## Yêu cầu hàm trả ra thứ tự các đỉnh được thăm (những đỉnh không thể thăm từ đỉnh 1 thì không phải in ra).



/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include<bits/stdc++.h>

using namespace std;

//hàm sử dụng thuật toán tìm kiếm theo chiều sâu

//adj là danh sách kề

void dfs(vector< list<int> > adj) {

//sử dụng stack để lưu các đỉnh được duyệt

stack<int> S;

vector<bool> visited(adj.size());

S.push(1); // Bắt đầu từ đỉnh số 1

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

//lặp lại các câu lệnh dưới đến khi stack rỗng

while (!S.empty()){

//đỉnh trên cùng là đỉnh hiện thời

int u = S.top();

//kiểm tra u đã được thăm hay chưa

if (!visited[u]){

//nếu chưa thăm thì đánh dấu nó

visited[u] = true;

//in ra giá trị của đỉnh đó

cout << u <<endl;

}

//kiểm tra xem đỉnh hiện thời có đỉnh kề được thăm chưa

if (!adj[u].empty()){

/\* nếu danh sách kề khác rỗng thì lấy đỉnh kề đầu tiên

của đỉnh hiện thời làm đỉnh kế \*/

int v = adj[u].front();

adj[u].pop\_front();

//kiểm tra đỉnh kế chưa được thăm thì đưa vào stack

if (!visited[v]) S.push(v);

}

//còn nếu danh sách kề rỗng thì loại bỏ đỉnh hiện thời đi

else S.pop();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

int main(){

int n = 7;

vector< list<int> > adj;

adj.resize(n + 1);

adj[1].push\_back(2);

adj[2].push\_back(4);

adj[1].push\_back(3);

adj[3].push\_back(4);

adj[3].push\_back(5);

adj[5].push\_back(2);

adj[2].push\_back(7);

adj[6].push\_back(7);

dfs(adj);

}

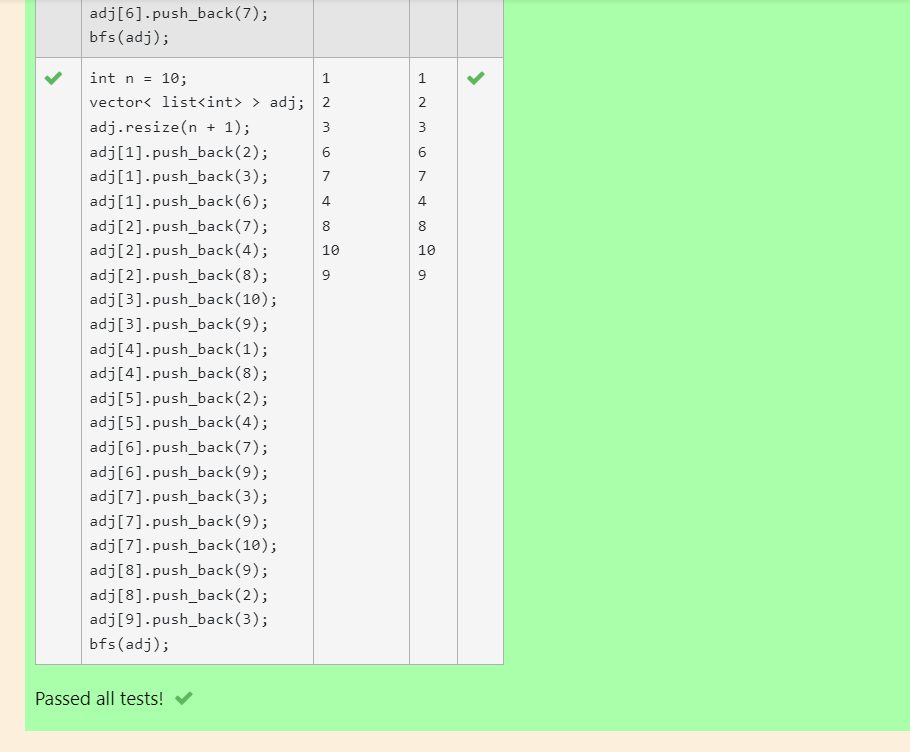
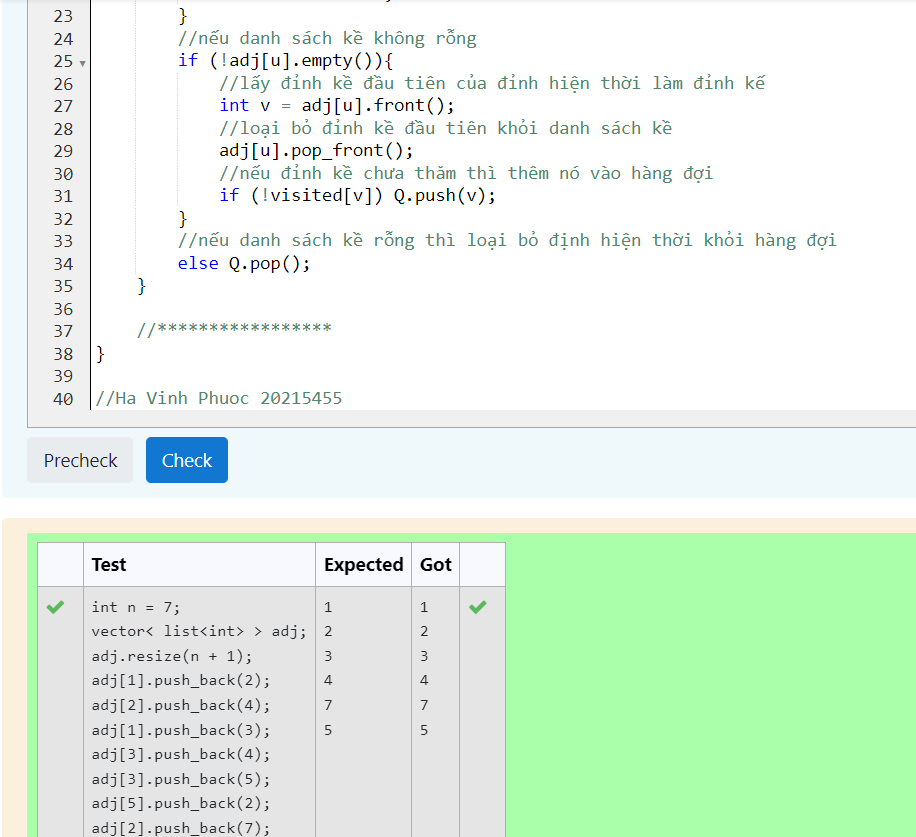
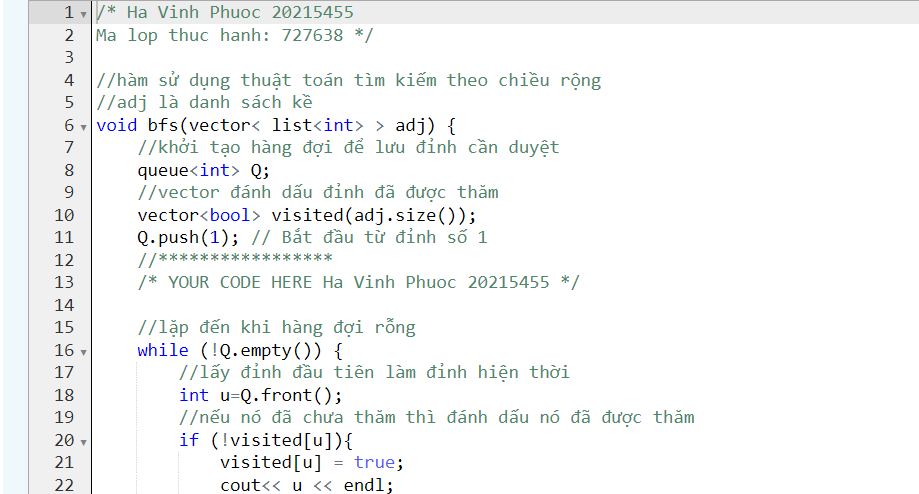
//Ha Vinh Phuoc 20215455

## **Bài tập 4.6.**

## Viết hàm thực hiện thuật toán BFS không sử dụng đệ quy trên đồ thị biểu diễn bằng danh sách kề vector< list<int> > .

## Đồ thị có n đỉnh được đánh số từ 1 đến n. Thuật toán BFS xuất phát từ đỉnh 1. Các đỉnh được thăm theo thứ tự ưu tiên từ trái sang phải trong danh sách kề.

## Yêu cầu hàm trả ra thứ tự các đỉnh được thăm (những đỉnh không thể thăm từ đỉnh 1 thì không phải in ra).



/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include<bits/stdc++.h>

using namespace std;

//hàm sử dụng thuật toán tìm kiếm theo chiều rộng

//adj là danh sách kề

void bfs(vector< list<int> > adj) {

//khởi tạo hàng đợi để lưu đỉnh cần duyệt

queue<int> Q;

//vector đánh dấu đỉnh đã được thăm

vector<bool> visited(adj.size());

Q.push(1); // Bắt đầu từ đỉnh số 1

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

//lặp đến khi hàng đợi rỗng

while (!Q.empty()) {

//lấy đỉnh đầu tiên làm đỉnh hiện thời

int u=Q.front();

//nếu nó đã chưa thăm thì đánh dấu nó đã được thăm

if (!visited[u]){

visited[u] = true;

cout<< u << endl;

}

//nếu danh sách kề không rỗng

if (!adj[u].empty()){

//lấy đỉnh kề đầu tiên của đỉnh hiện thời làm đỉnh kế

int v = adj[u].front();

//loại bỏ đỉnh kề đầu tiên khỏi danh sách kề

adj[u].pop\_front();

//nếu đỉnh kề chưa thăm thì thêm nó vào hàng đợi

if (!visited[v]) Q.push(v);

}

//nếu danh sách kề rỗng thì loại bỏ định hiện thời khỏi hàng đợi

else Q.pop();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

int main(){

int n = 7;

vector< list<int> > adj;

adj.resize(n + 1);

adj[1].push\_back(2);

adj[2].push\_back(4);

adj[1].push\_back(3);

adj[3].push\_back(4);

adj[3].push\_back(5);

adj[5].push\_back(2);

adj[2].push\_back(7);

adj[6].push\_back(7);

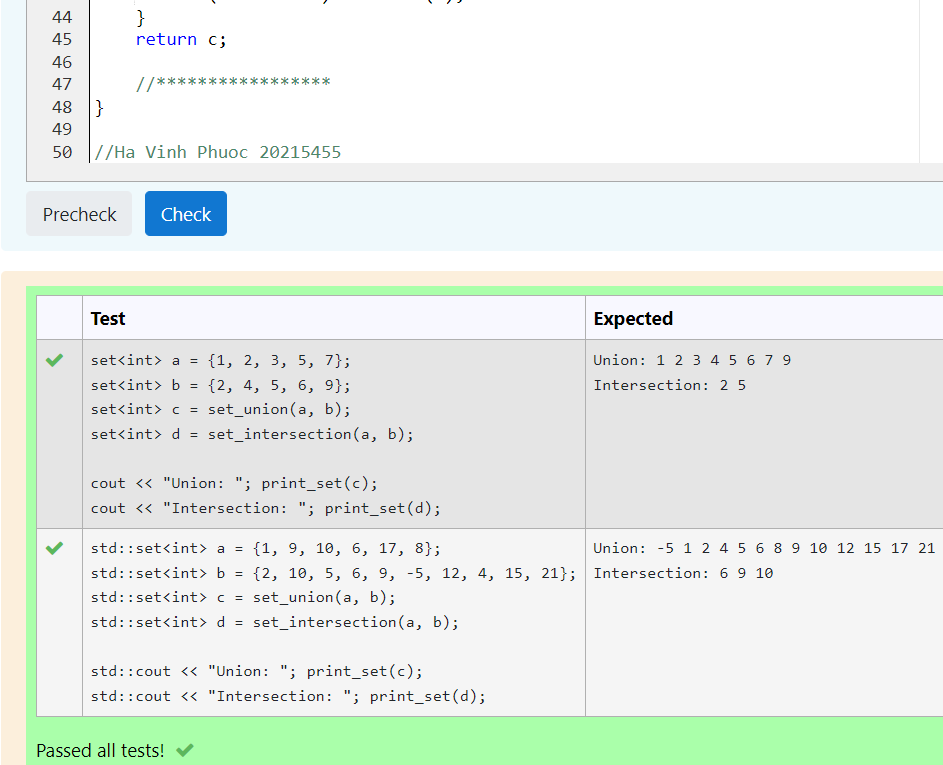
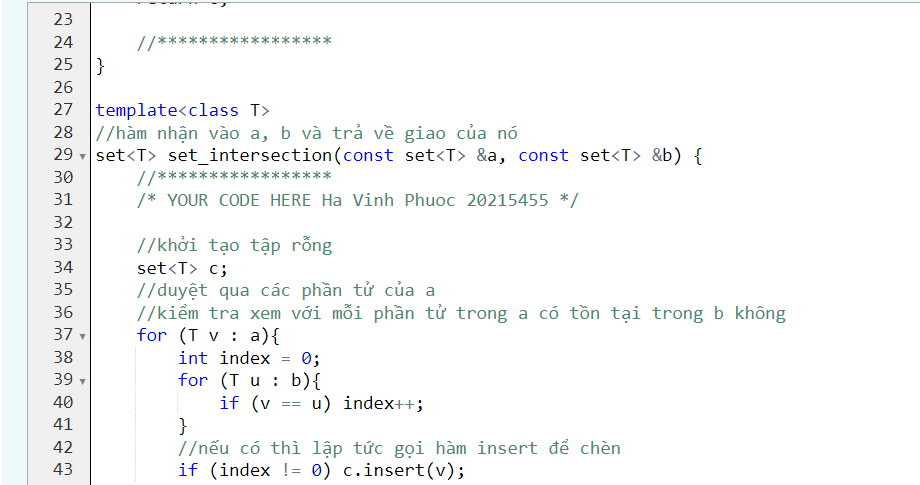
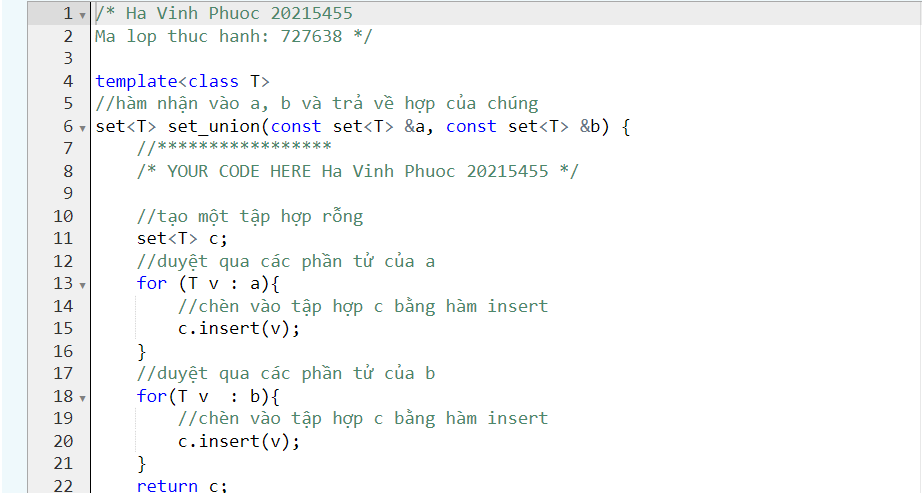
bfs(adj);

}

//Ha Vinh Phuoc 20215455

## **Bài tập 4.7.**

## Viết các hàm thực hiện các phép giao và hợp của hai tập hợp được biểu diễn bằng set.



/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include<iostream>

#include<set>

using namespace std;

template<class T>

//hàm nhận vào a, b và trả về hợp của chúng

set<T> set\_union(const set<T> &a, const set<T> &b) {

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

//tạo một tập hợp rỗng

set<T> c;

//duyệt qua các phần tử của a

for (T v : a){

//chèn vào tập hợp c bằng hàm insert

c.insert(v);

}

//duyệt qua các phần tử của b

for(T v : b){

//chèn vào tập hợp c bằng hàm insert

c.insert(v);

}

return c;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

template<class T>

//hàm nhận vào a, b và trả về giao của nó

set<T> set\_intersection(const set<T> &a, const set<T> &b) {

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

//khởi tạo tập rỗng

set<T> c;

//duyệt qua các phần tử của a

//kiểm tra xem với mỗi phần tử trong a có tồn tại trong b không

for (T v : a){

int index = 0;

for (T u : b){

if (v == u) index++;

}

//nếu có thì lập tức gọi hàm insert để chèn

if (index != 0) c.insert(v);

}

return c;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

template<class T>

//hàm dùng để in ra tất cả phần tử của tập hợp

void print\_set(const std::set<T> &a) {

for (const T &x : a) {

std::cout << x << ' ';

}

std::cout << std::endl;

}

int main() {

std::set<int> a = {1, 2, 3, 5, 7};

std::set<int> b = {2, 4, 5, 6, 9};

std::set<int> c = set\_union(a, b);

std::set<int> d = set\_intersection(a, b);

std::cout << "Union: "; print\_set(c);

std::cout << "Intersection: "; print\_set(d);

return 0;

}

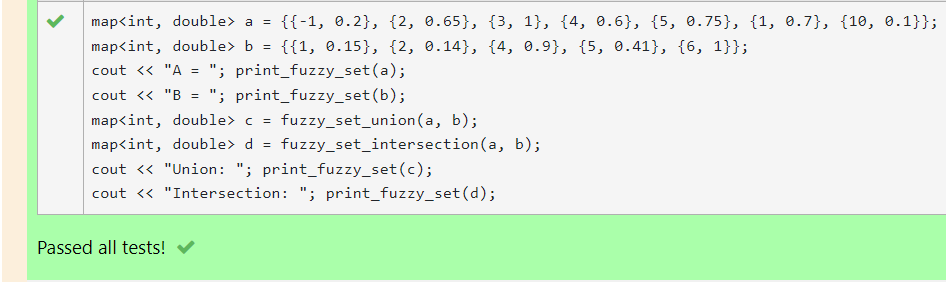
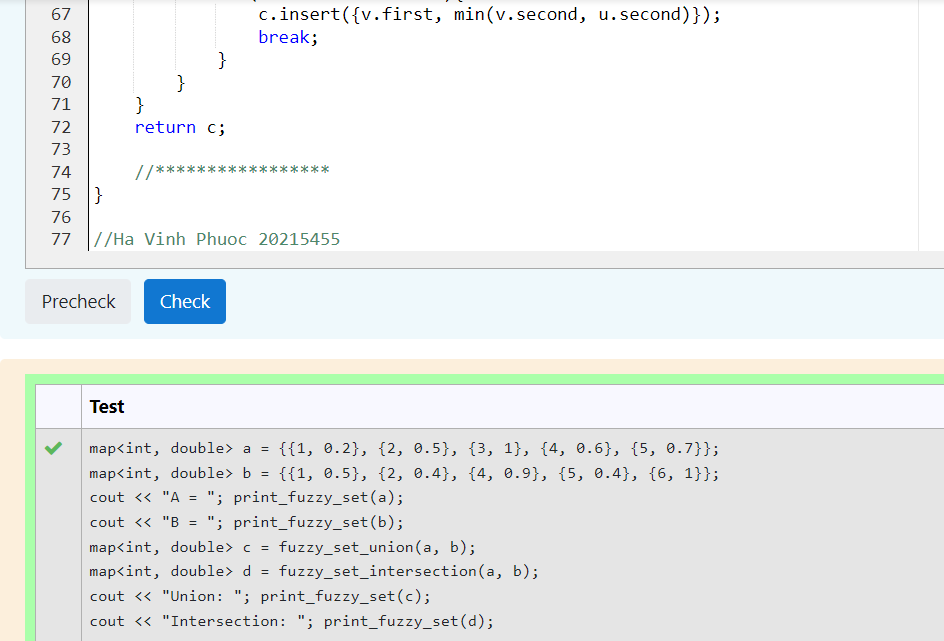
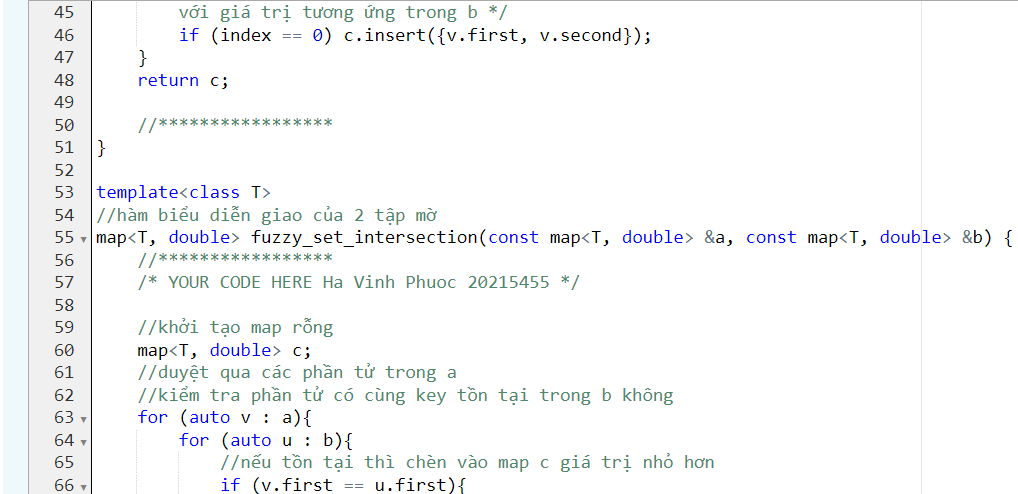
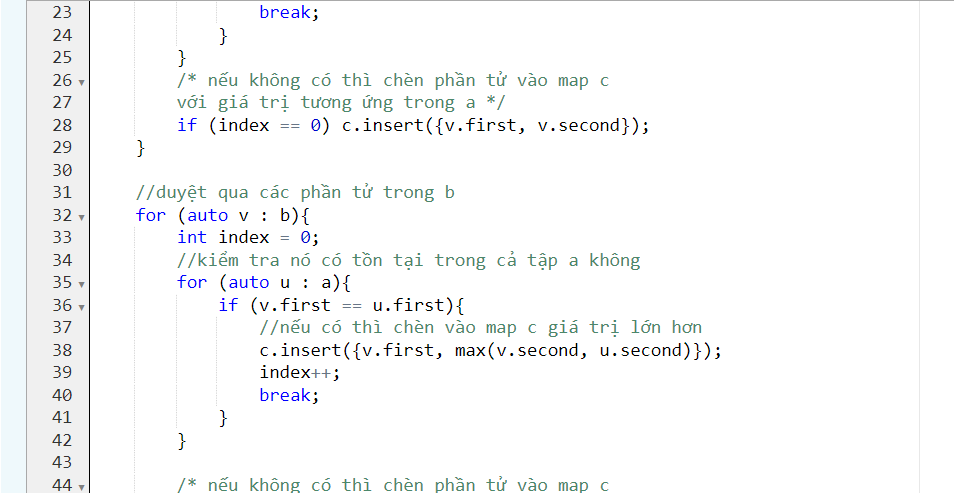
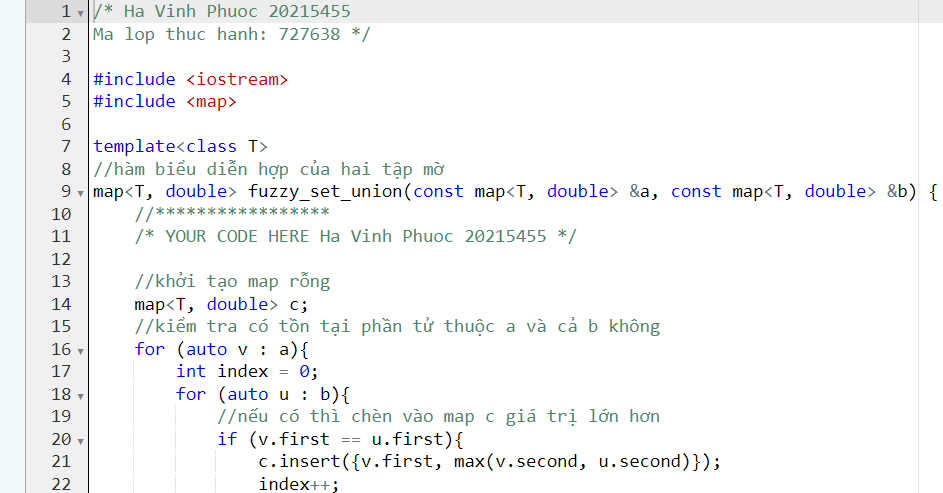
//Ha Vinh Phuoc 20215455

## **Bài tập 4.8.**

## Viết các hàm thực hiện các phép giao và hợp của hai tập hợp mờ được biểu diễn bằng map.

## Trong đó mỗi phần tử được gán cho một số thực trong đoạn [0..1] biểu thị độ thuộc của phần tử trong tập hợp, với độ thuộc bằng 1 nghĩa là phần tử chắc chắn thuộc vào tập hợp và ngược lại độ thuộc bằng 0 nghĩa là phần tử chắc chắn không thuộc trong tập hợp.

## Phép giao và hợp của 2 tập hợp được thực hiện trên các cặp phần tử bằng nhau của 2 tập hợp, với độ thuộc mới được tính bằng phép toán min và max của hai độ thuộc.



/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include <iostream>

#include <map>

using namespace std;

template<class T>

//hàm biểu diễn hợp của hai tập mờ

map<T, double> fuzzy\_set\_union(const map<T, double> &a, const map<T, double> &b) {

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

//khởi tạo map rỗng

map<T, double> c;

//kiểm tra có tồn tại phần tử thuộc a và cả b không

for (auto v : a){

int index = 0;

for (auto u : b){

//nếu có thì chèn vào map c giá trị lớn hơn

if (v.first == u.first){

c.insert({v.first, max(v.second, u.second)});

index++;

break;

}

}

/\* nếu không có thì chèn phần tử vào map c

với giá trị tương ứng trong a \*/

if (index == 0) c.insert({v.first, v.second});

}

//duyệt qua các phần tử trong b

for (auto v : b){

int index = 0;

//kiểm tra nó có tồn tại trong cả tập a không

for (auto u : a){

if (v.first == u.first){

//nếu có thì chèn vào map c giá trị lớn hơn

c.insert({v.first, max(v.second, u.second)});

index++;

break;

}

}

/\* nếu không có thì chèn phần tử vào map c

với giá trị tương ứng trong b \*/

if (index == 0) c.insert({v.first, v.second});

}

return c;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

template<class T>

//hàm biểu diễn giao của 2 tập mờ

map<T, double> fuzzy\_set\_intersection(const map<T, double> &a, const map<T, double> &b) {

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

//khởi tạo map rỗng

map<T, double> c;

//duyệt qua các phần tử trong a

//kiểm tra phần tử có cùng key tồn tại trong b không

for (auto v : a){

for (auto u : b){

//nếu tồn tại thì chèn vào map c giá trị nhỏ hơn

if (v.first == u.first){

c.insert({v.first, min(v.second, u.second)});

break;

}

}

}

return c;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

template<class T>

void print\_fuzzy\_set(const std::map<T, double> &a) {

cout << "{ ";

for (const auto &x : a) {

std::cout << "(" << x.first << ", " << x.second << ") ";

}

cout << "}";

std::cout << std::endl;

}

int main() {

std::map<int, double> a = {{1, 0.2}, {2, 0.5}, {3, 1}, {4, 0.6}, {5, 0.7}};

std::map<int, double> b = {{1, 0.5}, {2, 0.4}, {4, 0.9}, {5, 0.4}, {6, 1}};

std::cout << "A = "; print\_fuzzy\_set(a);

std::cout << "B = "; print\_fuzzy\_set(b);

std::map<int, double> c = fuzzy\_set\_union(a, b);

std::map<int, double> d = fuzzy\_set\_intersection(a, b);

std::cout << "Union: "; print\_fuzzy\_set(c);

std::cout << "Intersection: "; print\_fuzzy\_set(d);

}

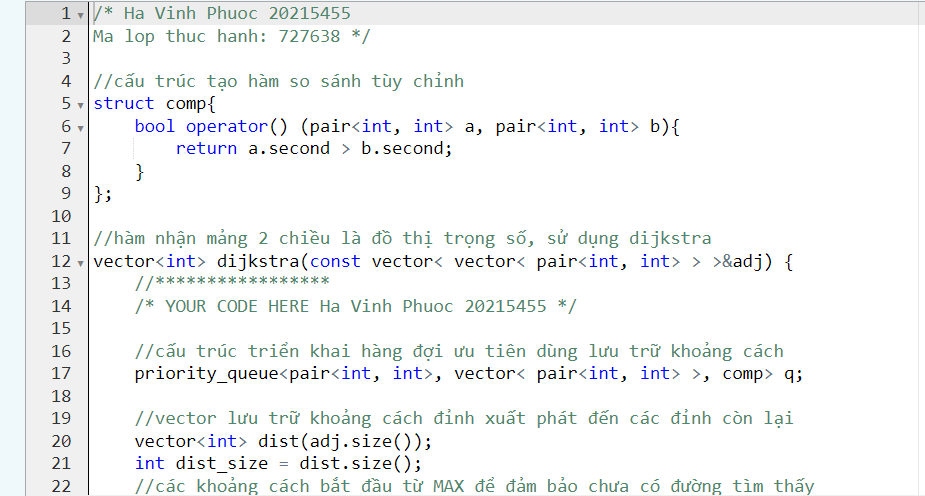
//Ha Vinh Phuoc 20215455

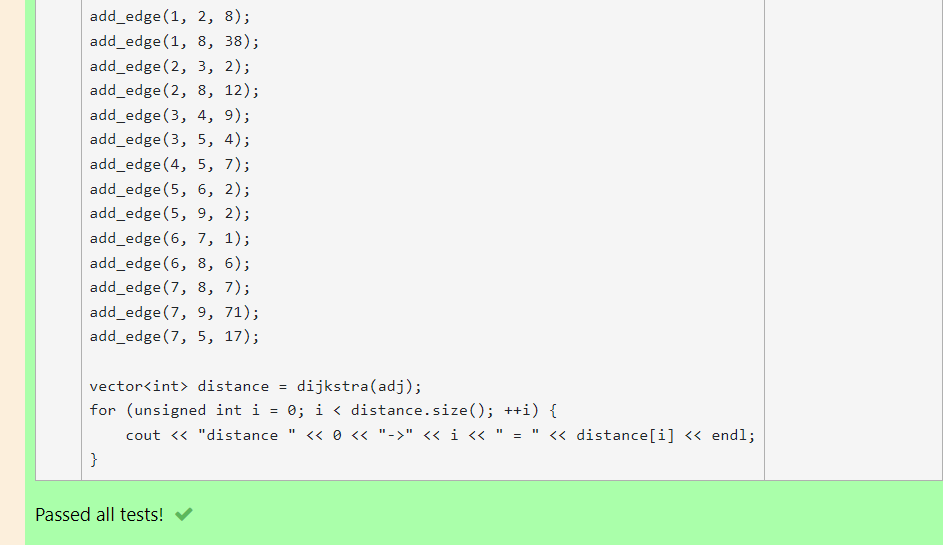
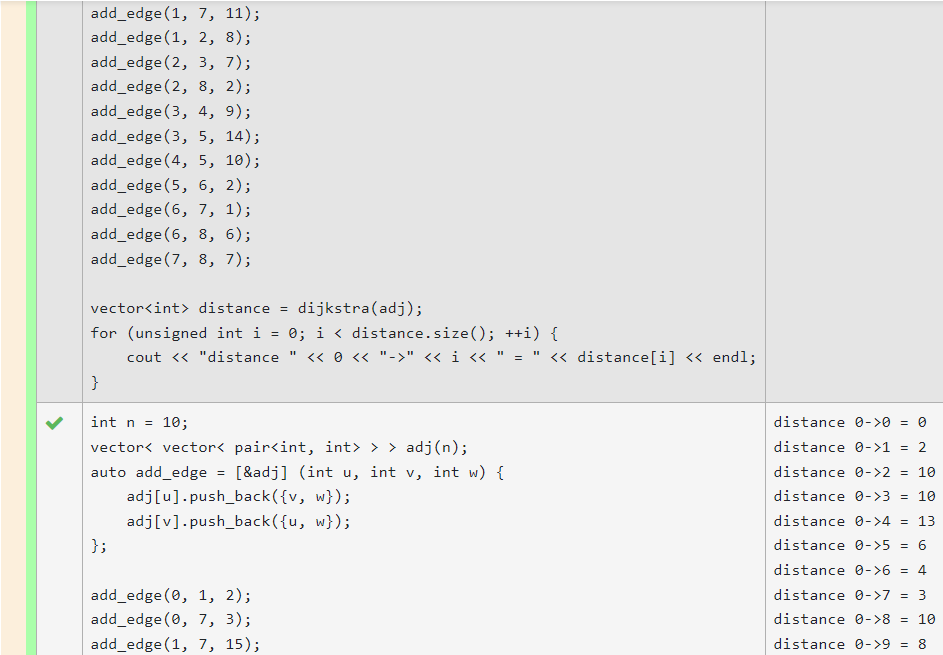
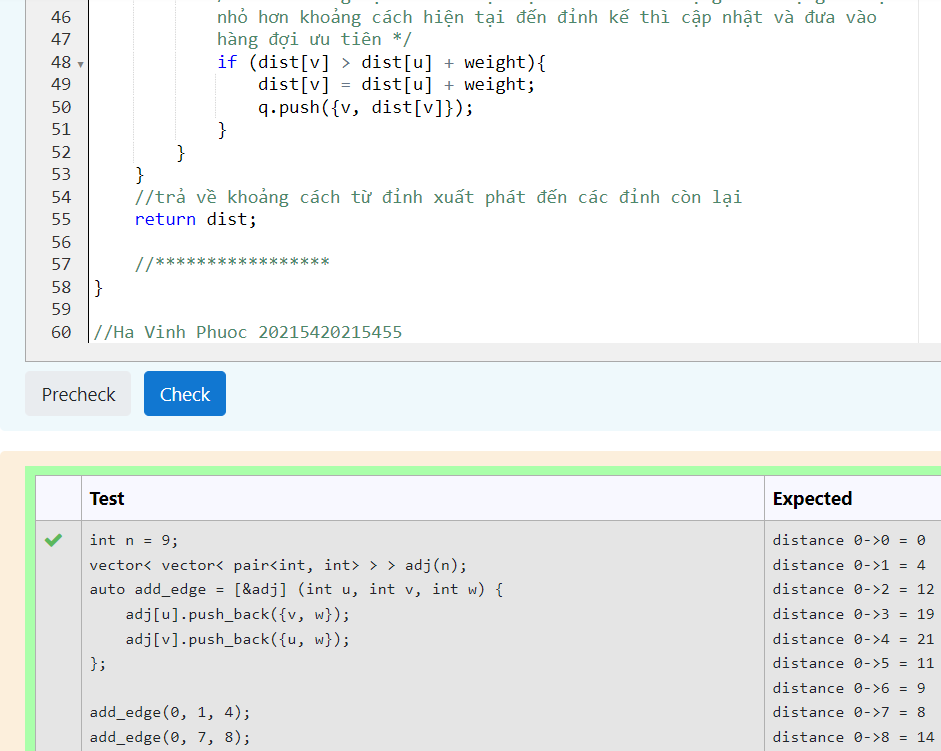
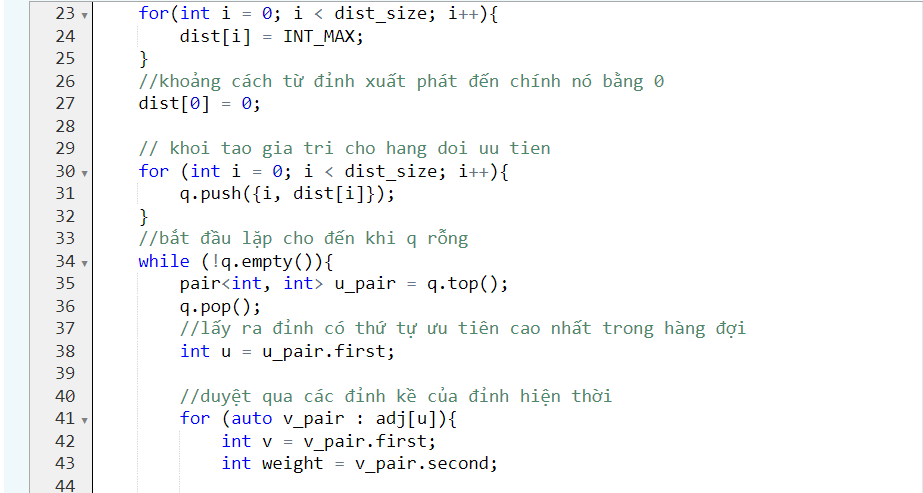
## **Bài tập 4.9.**

## Cài đặt thuật toán Dijkstra trên đồ thị vô hướng được biểu diễn bằng danh sách kề sử dụng priority\_queue

## Cụ thể, bạn cần cài đặt hàm vector<int> dijkstra(const vector< vector< pair<int, int> > >&adj) nhận đầu vào là danh sách kề chứa các cặp pair<int, int> biểu diễn đỉnh kề và trọng số tương ứng của cạnh. Đồ thị gồm n đỉnh được đánh số từ 0 tới n-1.

## Hàm cần trả vector<int> chứa n phần tử lần lượt là khoảng cách đường đi ngắn nhất từ đỉnh 0 tới các đỉnh 0, 1, 2, ..., n-1.





/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include <iostream>

#include <queue>

#include <climits>

using namespace std;

//cấu trúc tạo hàm so sánh tùy chỉnh

struct comp{

bool operator() (pair<int, int> a, pair<int, int> b){

return a.second > b.second;

}

};

//hàm nhận mảng 2 chiều là đồ thị trọng số, sử dụng dijkstra

vector<int> dijkstra(const vector< vector< pair<int, int> > >&adj) {

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* YOUR CODE HERE Ha Vinh Phuoc 20215455 \*/

//cấu trúc triển khai hàng đợi ưu tiên dùng lưu trữ khoảng cách

priority\_queue<pair<int, int>, vector< pair<int, int> >, comp> q;

//vector lưu trữ khoảng cách đỉnh xuất phát đến các đỉnh còn lại

vector<int> dist(adj.size());

int dist\_size = dist.size();

//các khoảng cách bắt đầu từ MAX để đảm bảo chưa có đường tìm thấy

for(int i = 0; i < dist\_size; i++){

dist[i] = INT\_MAX;

}

//khoảng cách từ đỉnh xuất phát đến chính nó bằng 0

dist[0] = 0;

// khoi tao gia tri cho hang doi uu tien

for (int i = 0; i < dist\_size; i++){

q.push({i, dist[i]});

}

//bắt đầu lặp cho đến khi q rỗng

while (!q.empty()){

pair<int, int> u\_pair = q.top();

q.pop();

//lấy ra đỉnh có thứ tự ưu tiên cao nhất trong hàng đợi

int u = u\_pair.first;

//duyệt qua các đỉnh kề của đỉnh hiện thời

for (auto v\_pair : adj[u]){

int v = v\_pair.first;

int weight = v\_pair.second;

/\* nếu khoảng cạnh đỉnh hiện tại đến đỉnh kể cộng với trọng số cạnh

nhỏ hơn khoảng cách hiện tại đến đỉnh kế thì cập nhật và đưa vào

hàng đợi ưu tiên \*/

if (dist[v] > dist[u] + weight){

dist[v] = dist[u] + weight;

q.push({v, dist[v]});

}

}

}

//trả về khoảng cách từ đỉnh xuất phát đến các đỉnh còn lại

return dist;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

int main() {

int n = 9;

vector< vector< pair<int, int> > > adj(n);

auto add\_edge = [&adj] (int u, int v, int w) {

adj[u].push\_back({v, w});

adj[v].push\_back({u, w});

};

add\_edge(0, 1, 4);

add\_edge(0, 7, 8);

add\_edge(1, 7, 11);

add\_edge(1, 2, 8);

add\_edge(2, 3, 7);

add\_edge(2, 8, 2);

add\_edge(3, 4, 9);

add\_edge(3, 5, 14);

add\_edge(4, 5, 10);

add\_edge(5, 6, 2);

add\_edge(6, 7, 1);

add\_edge(6, 8, 6);

add\_edge(7, 8, 7);

vector<int> distance = dijkstra(adj);

for (int i = 0; i < distance.size(); ++i) {

cout << "distance " << 0 << "->" << i << " = " << distance[i] << endl;

}

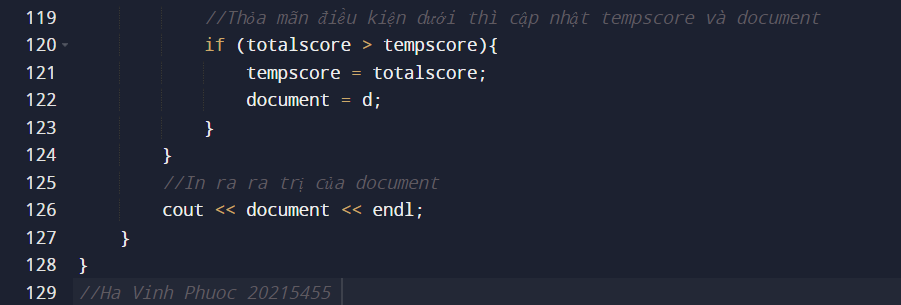
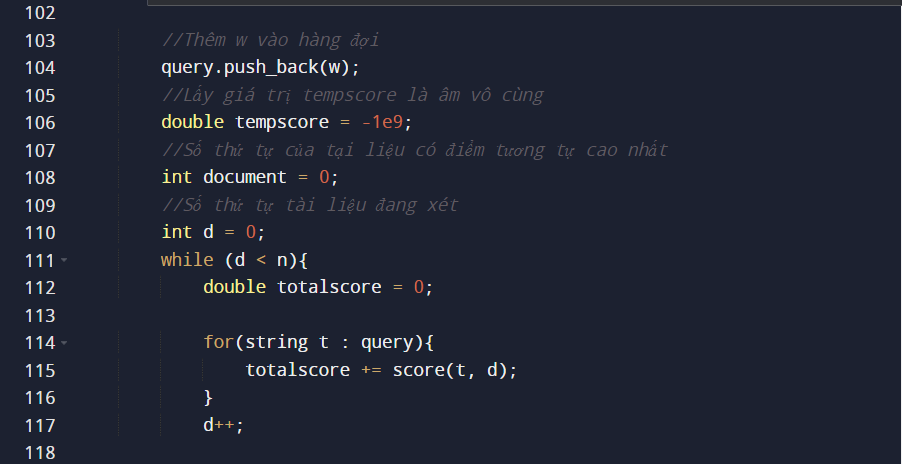
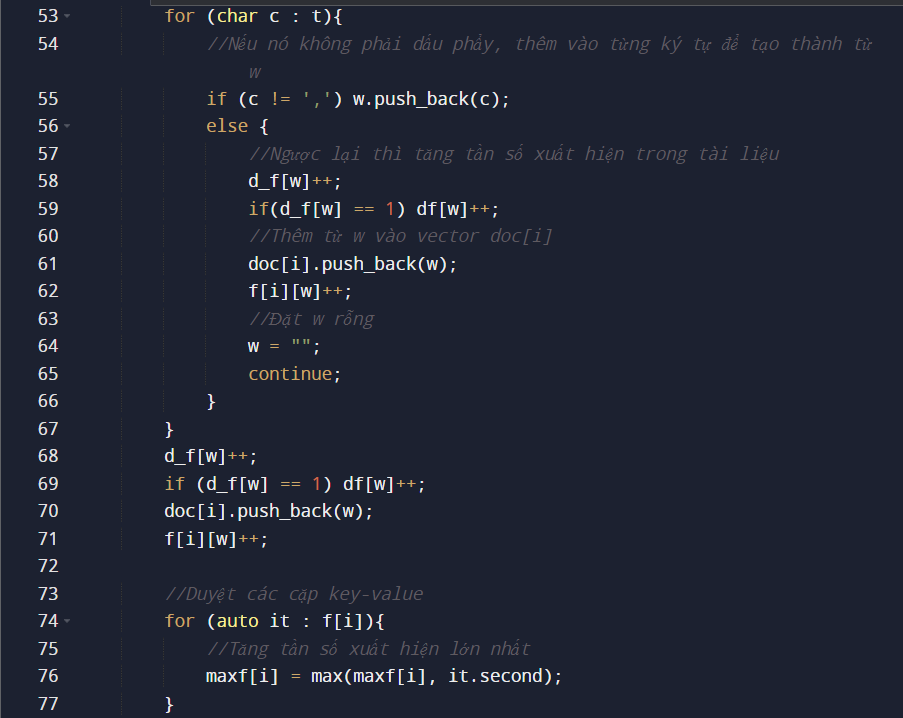
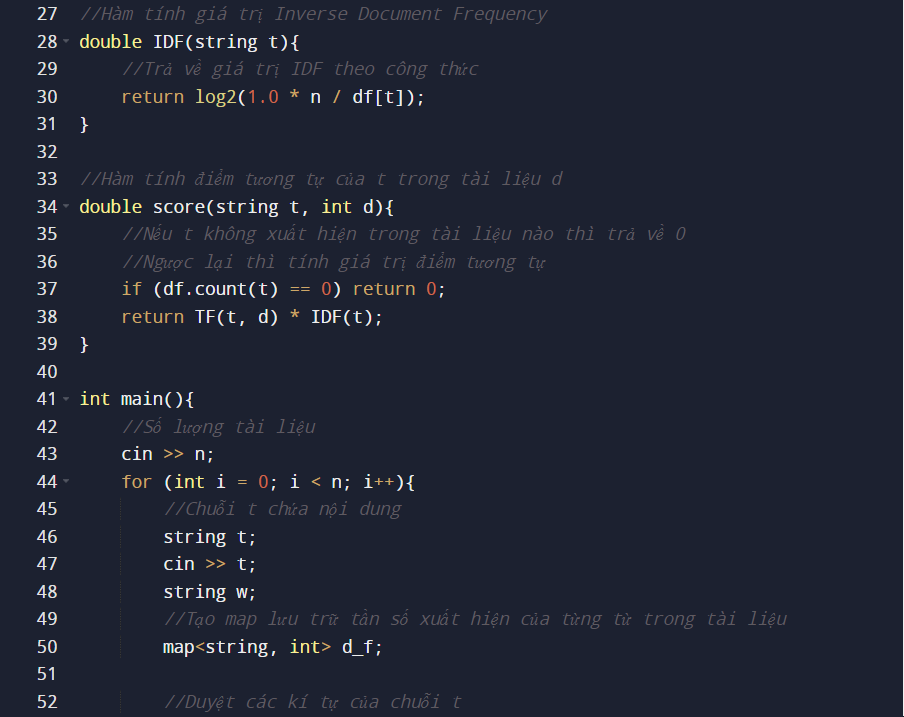
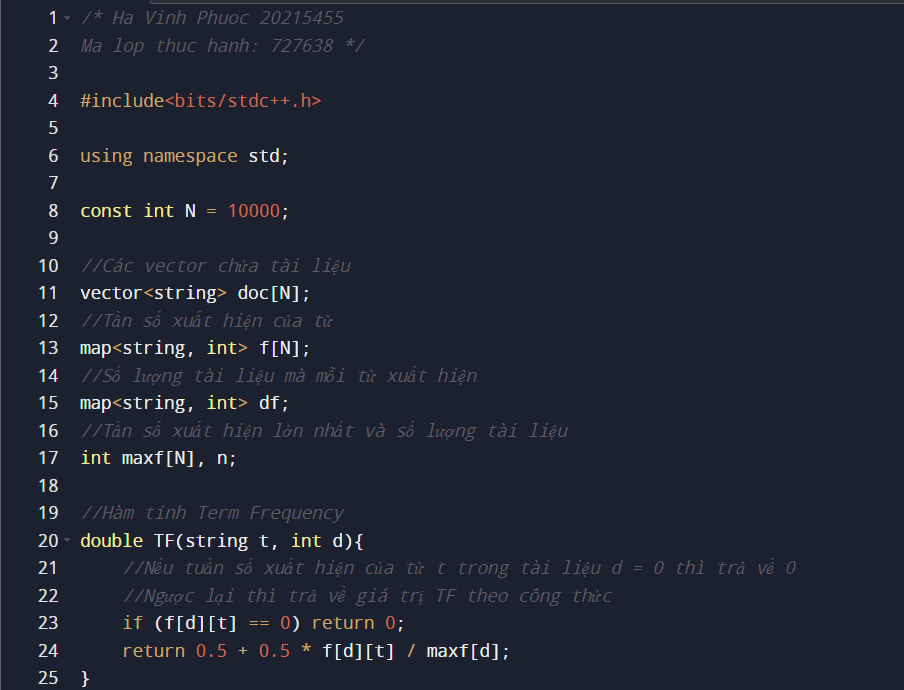
return 0;

}

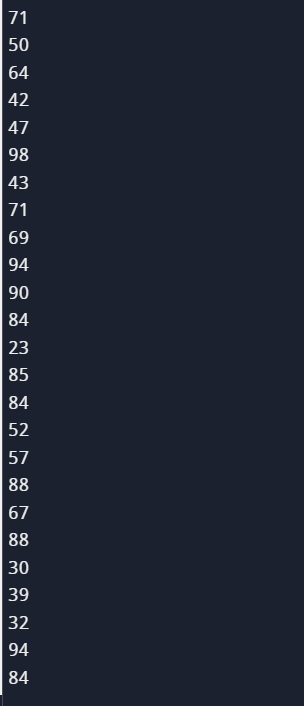
//Ha Vinh Phuoc 20215420215455

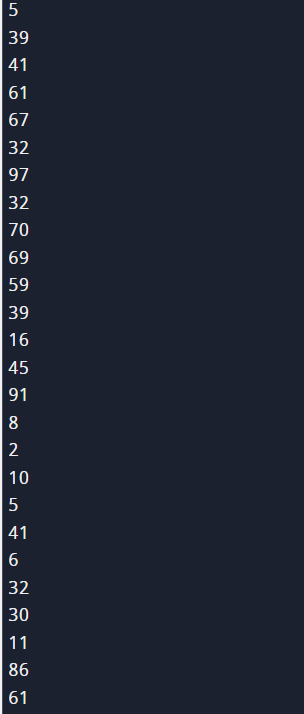
## **Bài tập 4.10. Search Engine**

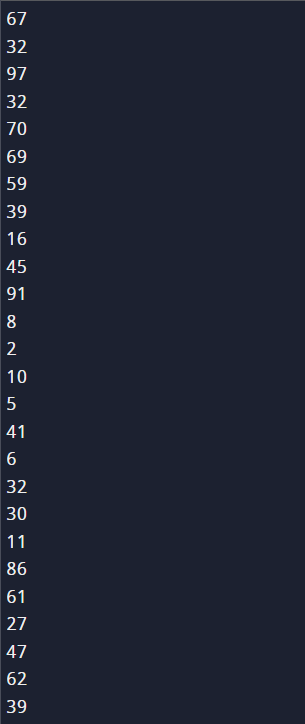
## Xây dựng một máy tìm kiếm (search engine) đơn giản.

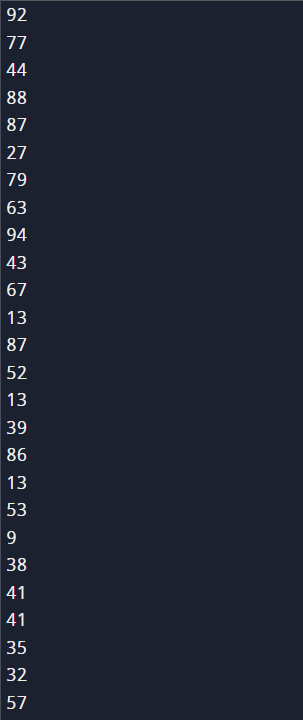


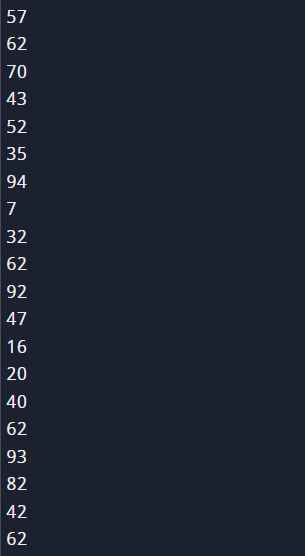
**Case = 1:**



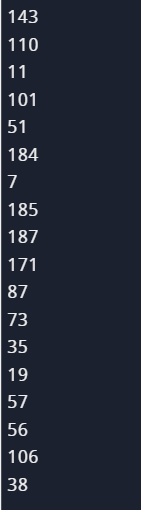
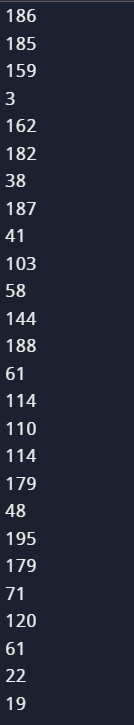
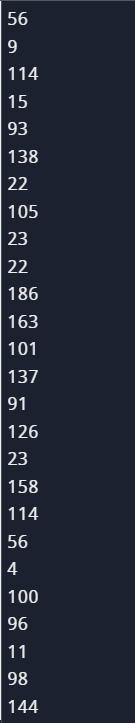
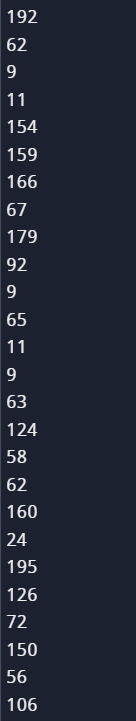
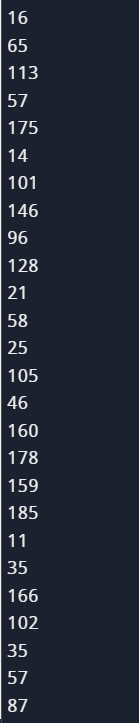
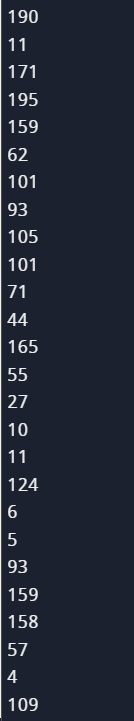
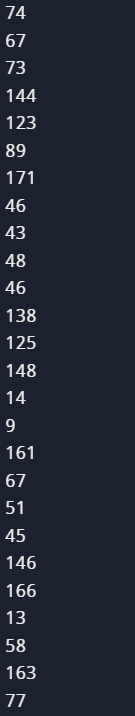
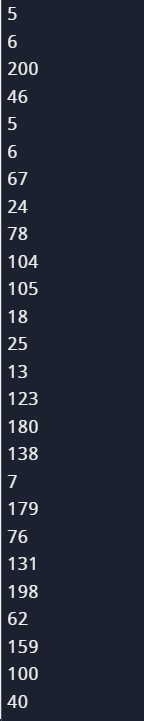




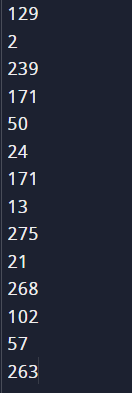
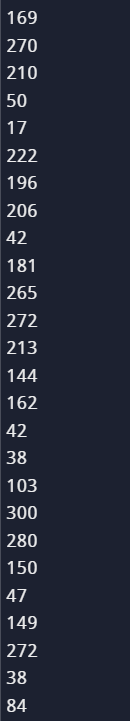
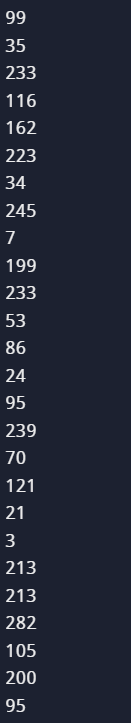
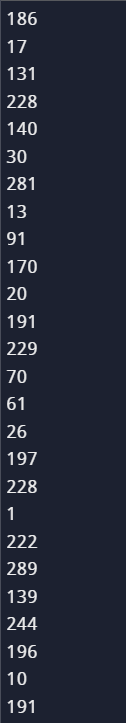
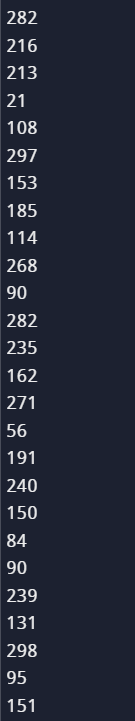
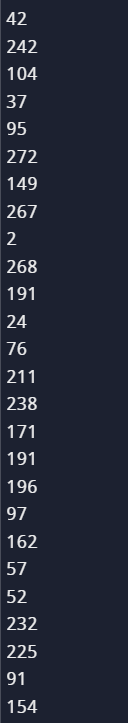
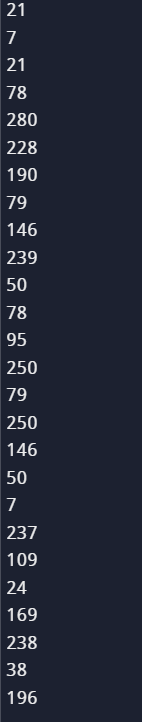
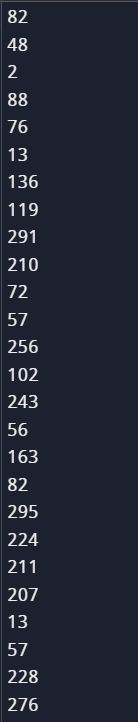
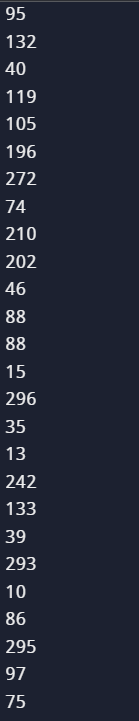
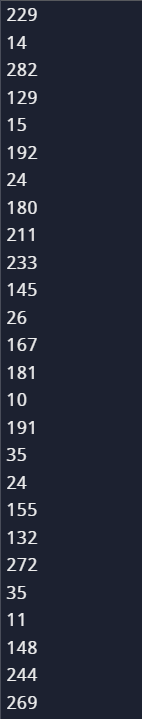
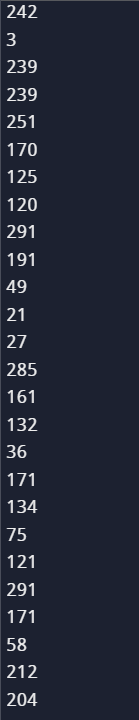
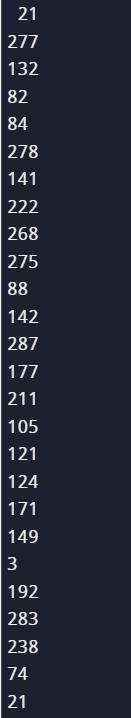




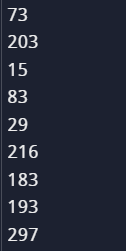
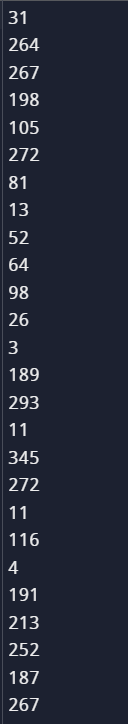
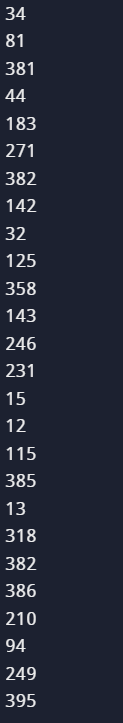
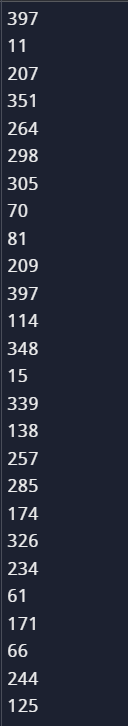
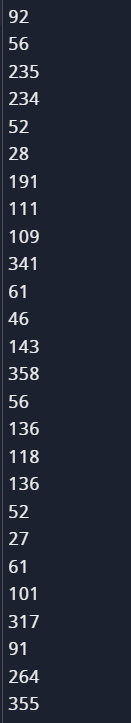
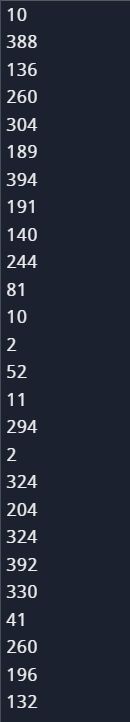
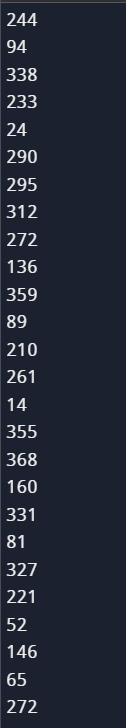
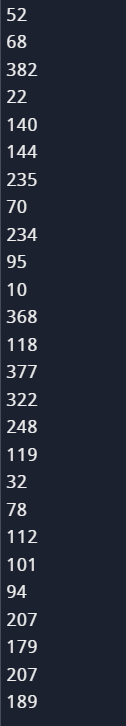
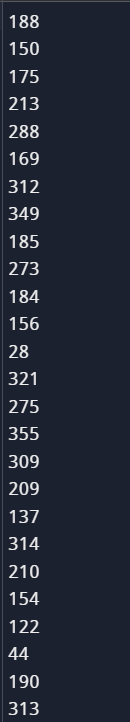
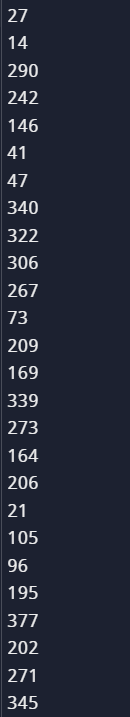
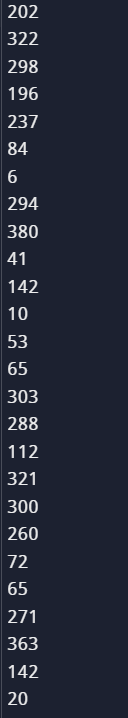
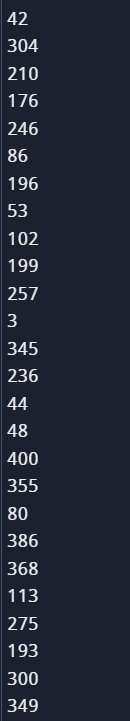
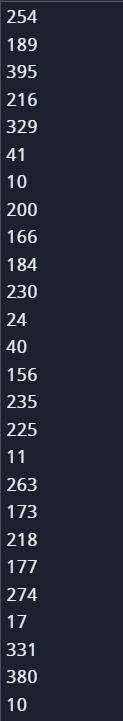
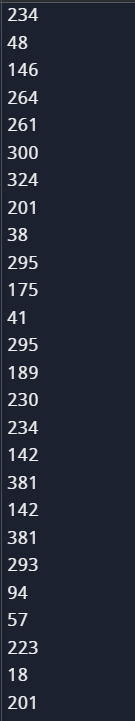
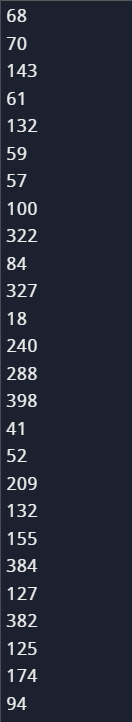
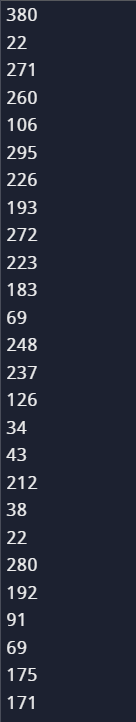
**Case = 2:**



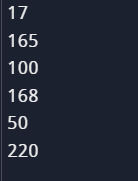
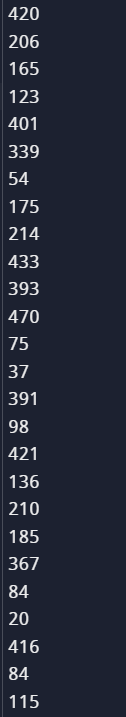
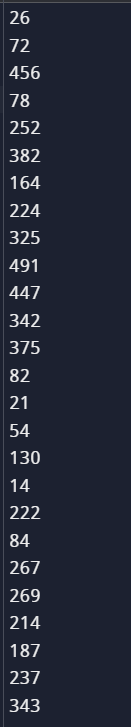
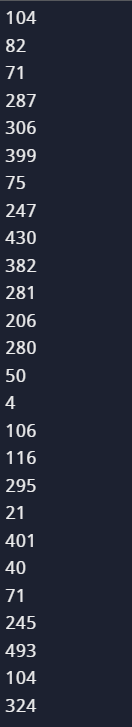
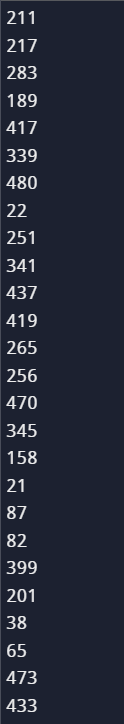
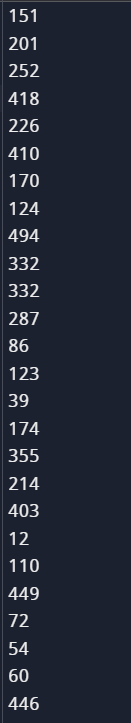
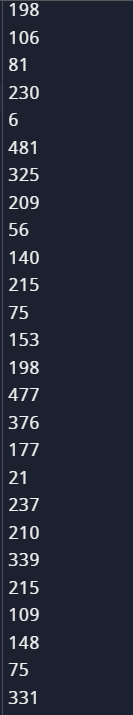
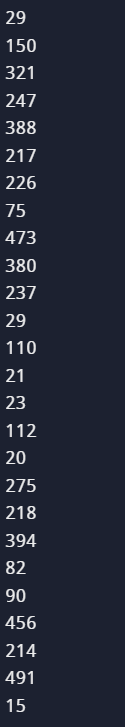
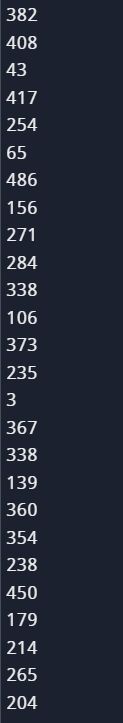
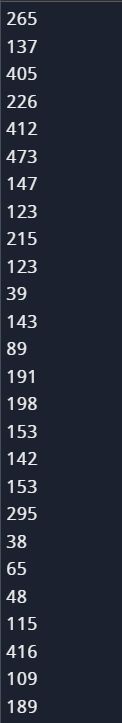
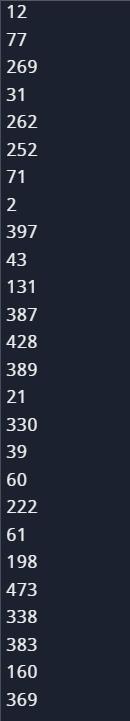
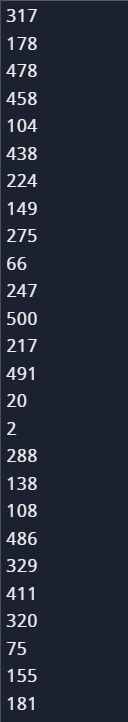
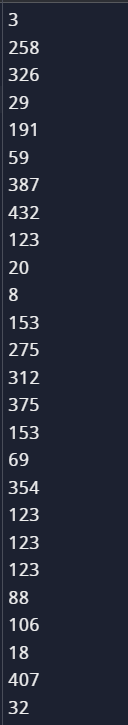
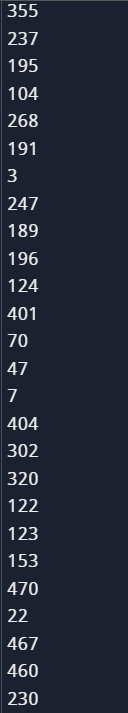
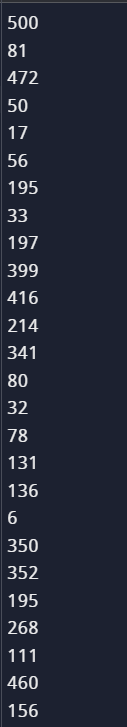
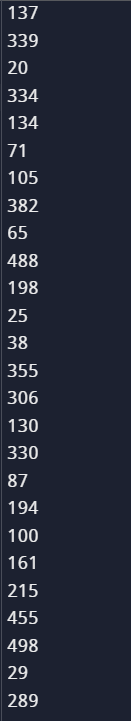
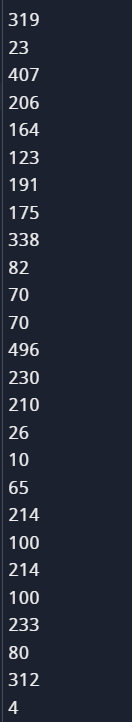
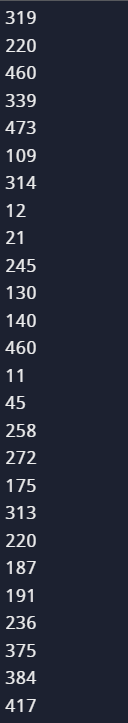
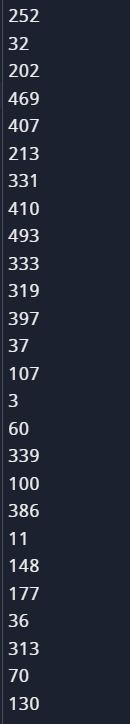
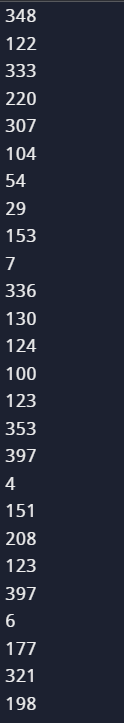
**Case = 3:**



**Case = 4:**



**Case = 5:**



## /\* Ha Vinh Phuoc 20215455

## Ma lop thuc hanh: 727638 \*/

## #include<bits/stdc++.h>

## using namespace std;

## const int N = 10000;

## //Các vector chứa tài liệu

## vector<string> doc[N];

## //Tần số xuất hiện của từ

## map<string, int> f[N];

## //Số lượng tài liệu mà mỗi từ xuất hiện

## map<string, int> df;

## //Tần số xuất hiện lớn nhất và số lượng tài liệu

## int maxf[N], n;

## //Hàm tính Term Frequency

## double TF(string t, int d){

## //Nếu tuần số xuất hiện của từ t trong tài liệu d = 0 thì trả về 0

## //Ngược lại thì trả về giá trị TF theo công thức

## if (f[d][t] == 0) return 0;

## return 0.5 + 0.5 \* f[d][t] / maxf[d];

## }

## //Hàm tính giá trị Inverse Document Frequency

## double IDF(string t){

## //Trả về giá trị IDF theo công thức

## return log2(1.0 \* n / df[t]);

## }

## //Hàm tính điểm tương tự của t trong tài liệu d

## double score(string t, int d){

## //Nếu t không xuất hiện trong tài liệu nào thì trả về 0

## //Ngược lại thì tính giá trị điểm tương tự

## if (df.count(t) == 0) return 0;

## return TF(t, d) \* IDF(t);

## }

## int main(){

## //Số lượng tài liệu

## cin >> n;

## for (int i = 0; i < n; i++){

## //Chuỗi t chứa nội dung

## string t;

## cin >> t;

## string w;

## //Tạo map lưu trữ tần số xuất hiện của từng từ trong tài liệu

## map<string, int> d\_f;

## //Duyệt các kí tự của chuỗi t

## for (char c : t){

## //Nếu nó không phải dấu phẩy, thêm vào từng ký tự để tạo thành từ w

## if (c != ',') w.push\_back(c);

## else {

## //Ngược lại thì tăng tần số xuất hiện trong tài liệu

## d\_f[w]++;

## if(d\_f[w] == 1) df[w]++;

## //Thêm từ w vào vector doc[i]

## doc[i].push\_back(w);

## f[i][w]++;

## //Đặt w rỗng

## w = "";

## continue;

## }

## }

## d\_f[w]++;

## if (d\_f[w] == 1) df[w]++;

## doc[i].push\_back(w);

## f[i][w]++;

## //Duyệt các cặp key-value

## for (auto it : f[i]){

## //Tăng tần số xuất hiện lớn nhất

## maxf[i] = max(maxf[i], it.second);

## }

## }

## //Số lượng câu truy vấn

## int q;

## cin >> q;

## while (q--){

## //Tạo vector lưu trữ từng từ trong câu truy vấn

## vector<string> query;

## //Chuỗi chứa nội dung câu truy vấn

## string t;

## cin >> t;

## string w;

## //Duyệt qua các ký tự trong chuỗi t

## for (char c : t){

## //Nếu ký tự không là dấu phẩy thì thêm từng ký tự để tạo thành từ w

## if(c != ',') w.push\_back(c);

## //Ngược lại thì thêm w vào hàng đợi và đặt w về rỗng để xử lý từ sau

## else{

## query.push\_back(w);

## w = "";

## continue;

## }

## }

## //Thêm w vào hàng đợi

## query.push\_back(w);

## //Lấy giá trị tempscore là âm vô cùng

## double tempscore = -1e9;

## //Số thứ tự của tại liệu có điểm tương tự cao nhất

## int document = 0;

## //Số thứ tự tài liệu đang xét

## int d = 0;

## while (d < n){

## double totalscore = 0;

## for(string t : query){

## totalscore += score(t, d);

## }

## d++;

## //Thỏa mãn điều kiện dưới thì cập nhật tempscore và document

## if (totalscore > tempscore){

## tempscore = totalscore;

## document = d;

## }

## }

## //In ra ra trị của document

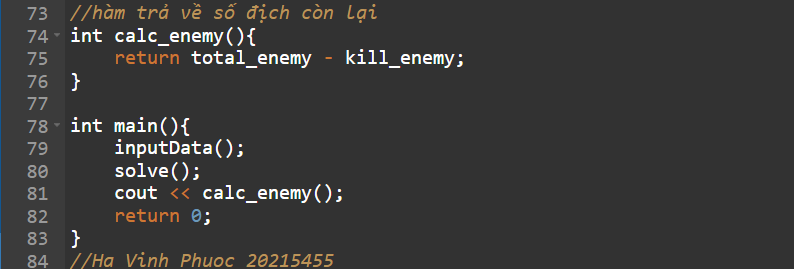
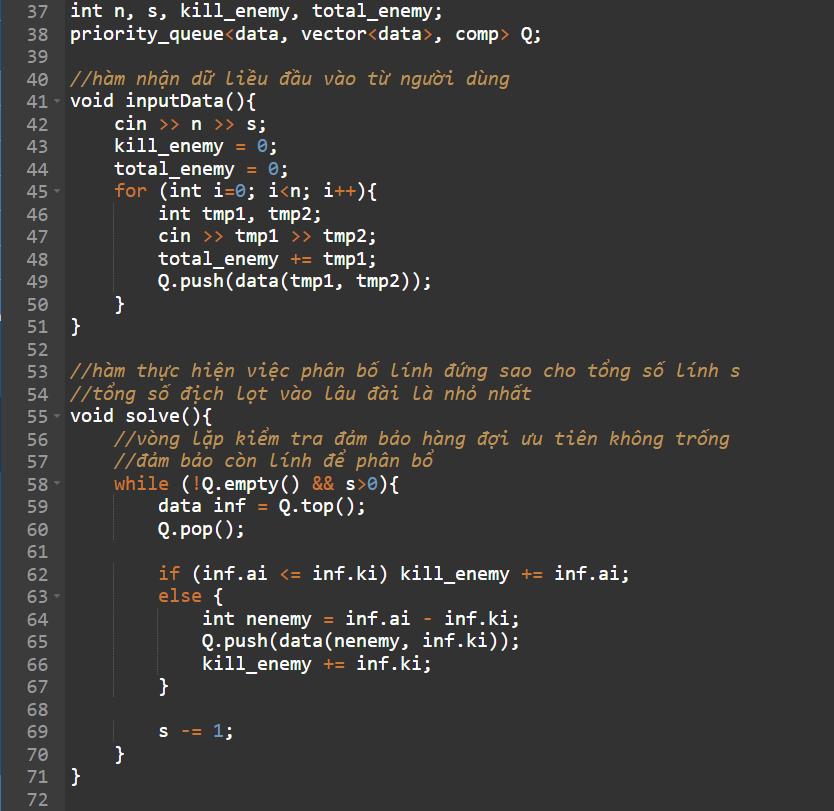
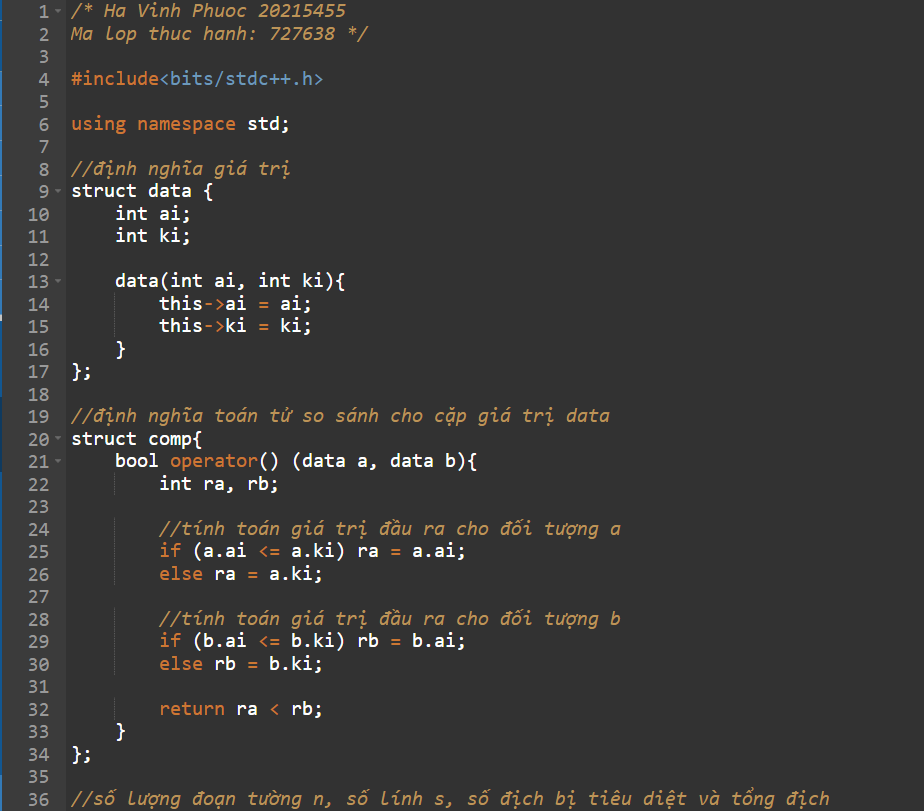
## cout << document << endl;

## }

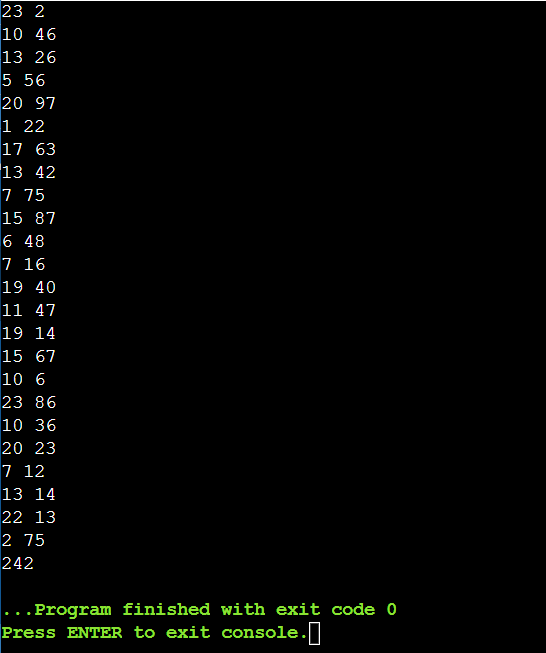
## }

## //Ha Vinh Phuoc 20215455

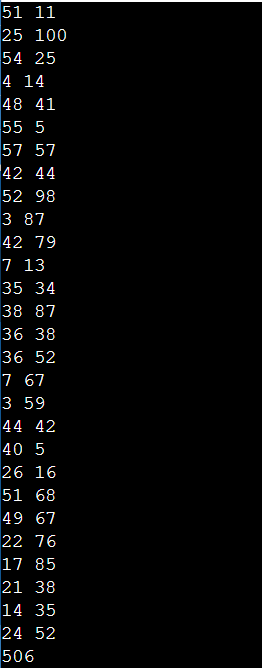
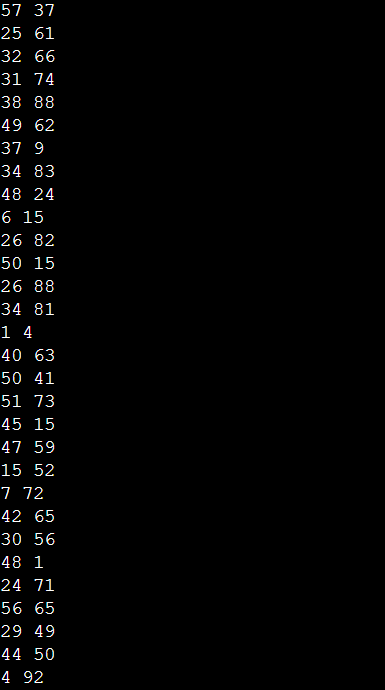
## **Bài tập 4.11. Bảo vệ lâu đài**



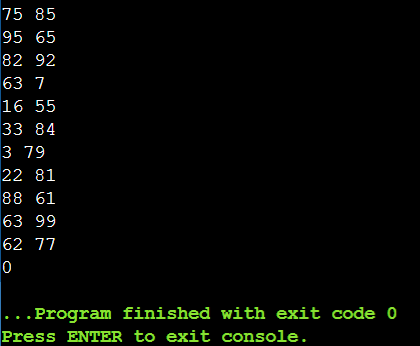
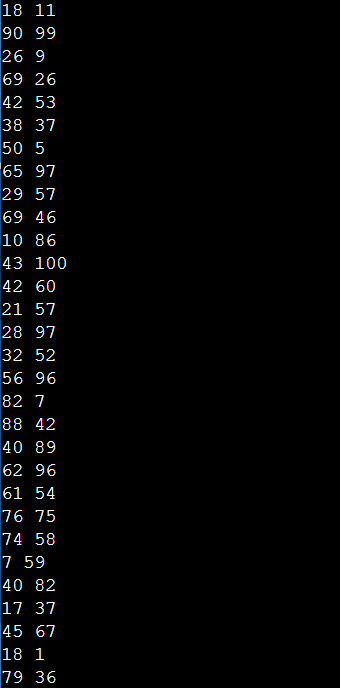
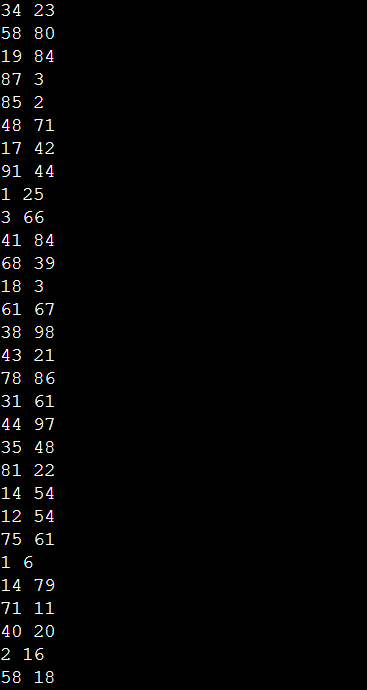
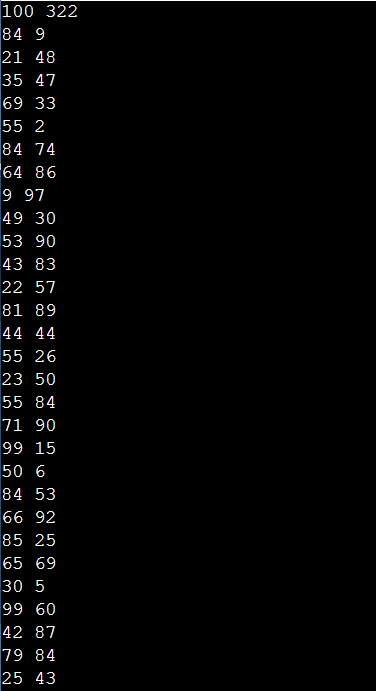
**Case = 1:**



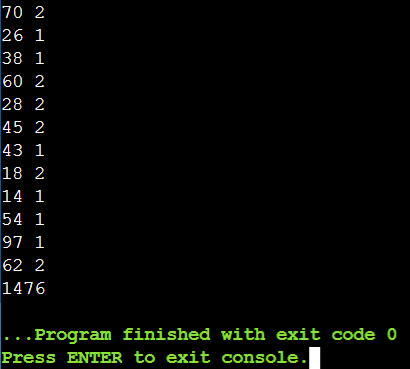
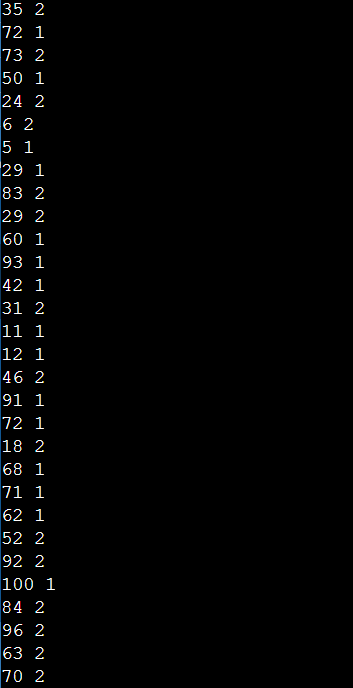
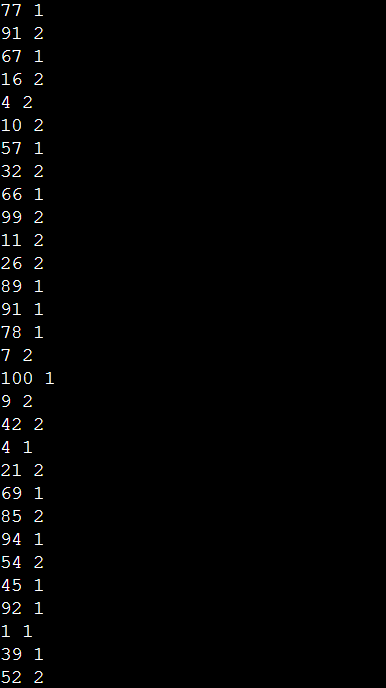
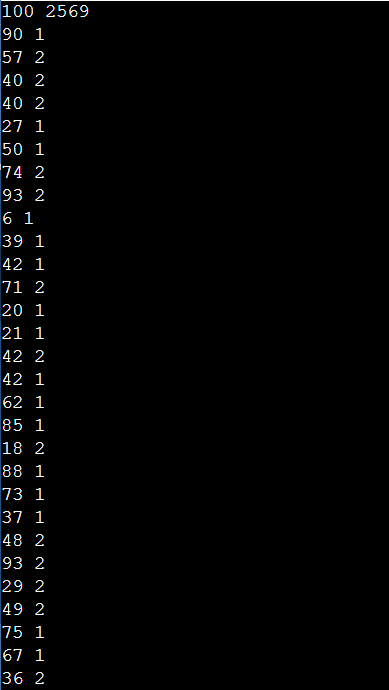
**Case = 2:**



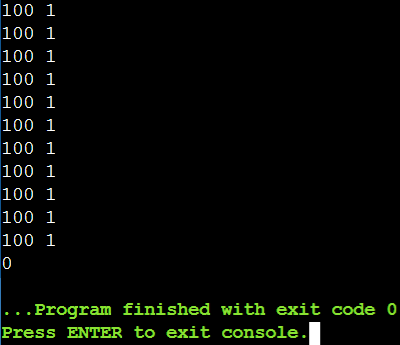
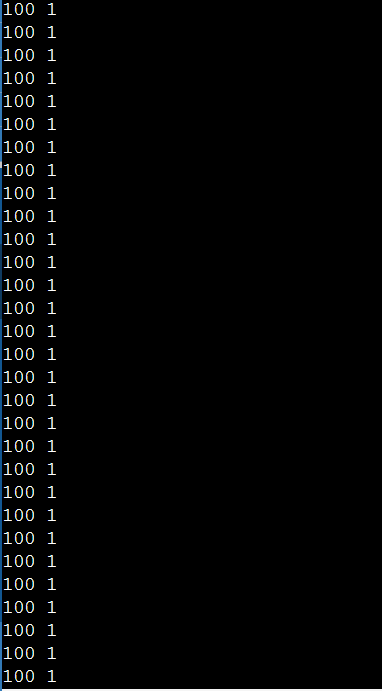
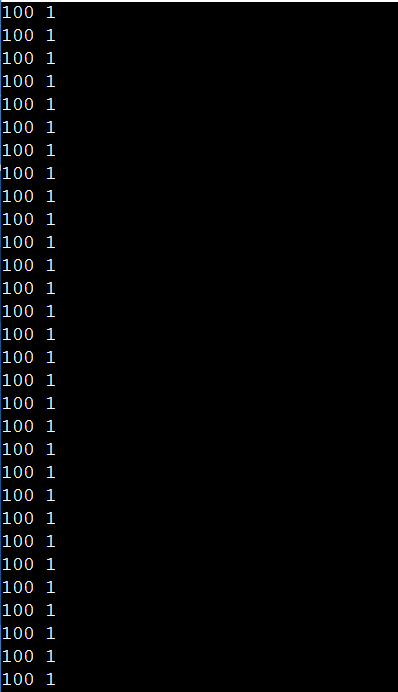
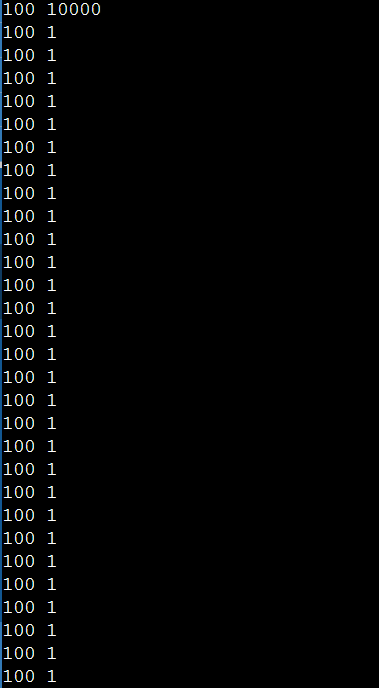
**Case = 3:**



**Case = 4:**

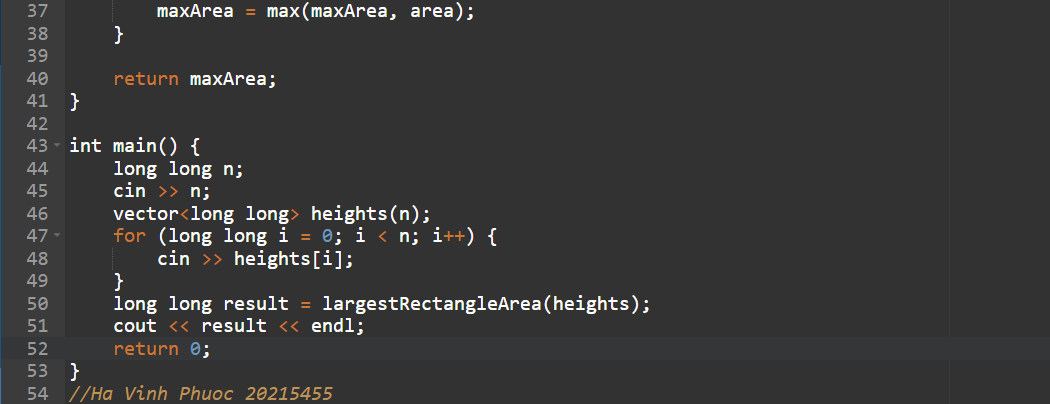
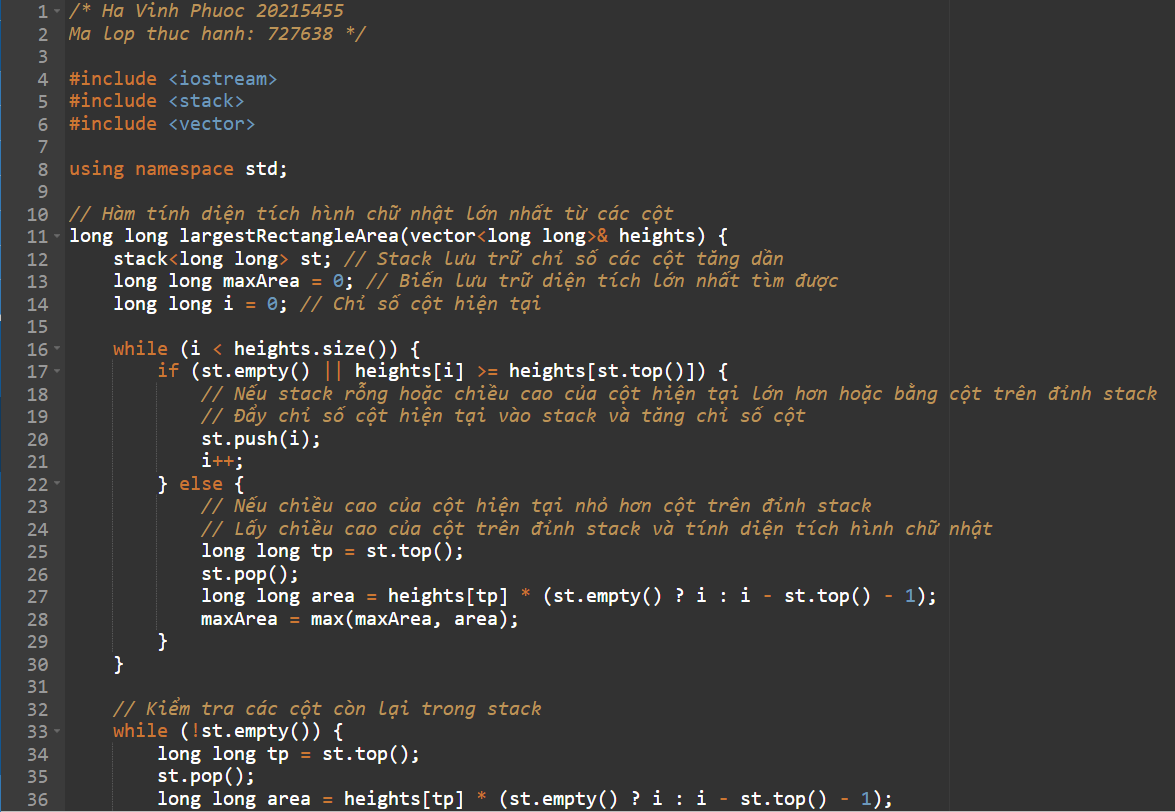


**Case = 5:**



## **Bài tập 4.12. Lược đồ**

## Cho một lược đồ gồm n cột chữ nhật liên tiếp nhau có chiều rộng bằng 1 và chiều cao lần lượt là các số nguyên không âm h1,h2,…,hn . Hãy xác định hình chữ nhật có diện tích lớn nhất có thể tạo thành từ các cột liên tiếp.



**Case = 1:**

10000

3368 3433 2649 2068 2548 3618 779 231 1185 4459 331 1938 4244 724 287 1048 3033 1205 441 1774 3379 3968 4560 742 596 1472 4221 1675 866 1907 585 383 732 1732 4621 3570 3771 508 3059 316 4644 3882 1019 4108 1540 4840 391 3833 3477 1328 2587 1969 1945 2399 352 1953 4076 97 3976 4485 2814 2989 2567 2512 2145 1410 3568 901 2249 242 4589 3135 2898 990 2883 2823 3542 2459 2053 1292 798 1523 264 2807 1888 1246 3471 455 547 1763 3471 4162 4199 3858 3046 1742 2248 4508 2096 4967 4621 20 4194 409 1211 3808 3971 2665 3528 3627 2783 4615 149 2293 150 4134 1323 4902 5 4567 4937 1574 1747 4547 84 368 3182 16 3937 2760 3726 94 3239 2587 1299 2594 4177 2822 1547 35 2186 4474 159 963 473 1070 3678 1456 232 2230 2178 1883 967 2098 1522 2368 1782 3922 4735 248 1131 4296 1731 4659 4024 1766 2622 3882 4641 450 2091 1989 4845 1672 2355 1768 3435 2215 425 4859 4311 1478 2278 961 77 600 2501 4473 1940 1461 2297 1852 4311 1940 829 2694 840 4157 3789 4969 942 3268 764 4630 2009 2435 3149 1661 27 4247 254 4551 593 902 2657 4098 3570 3503 836 61 2847 475 3292 213 1676 442 4958 1119 4958 209 4415 2518 3510 2547 4708 3024 1053 2293 1380 4548 3960 3150 1172 3094 1286 4839 4571 630 2703 4159 2432 3990 2503 4531 744 2755 3523 1484 4501 3670 4523 3145 3555 235 2729 1487 2517 1281 4530 85 951 835 3274 4069 3173 3703 3237 400 3268 3093 2523 2849 1335 162 2475 2741 2742 2334 2331 3673 4057 4738 2319 1733 4259 22 544 4680 2533 1402 4095 2964 2440 1899 3731 4075 1000 4177 702 3545 669 2759 4103 4548 1347 3138 43 2530 2972 3845 2677 314 2444 1068 4073 822 3478 288 2691 2487 1246 507 4349 857 2646 2852 416 2168 308 205 2579 3011 988 171 4538 624 2714 3134 2129 4097 723 921 1857 3870 2693 4524 1849 3035 3228 3177 4857 587 2662 835 1763 2004 2273 3078 3908 2045 3203 2595 170 309 3506 1842 4717 446 2670 4413 1986 1490 2580 3163 1494 3733 2052 4322 1192 4844 2821 2336 2595 428 4471 4748 218 3493 4857 2928 805 1560 4308 1738 4419 1336 872 3556 1016 4237 2126 1897 4121 4302 1261 1672 3701 3992 3681 2566 1214 497 3241 2919 1408 4850 808 1956 1002 4653 1766 4260 3731 3268 4159 916 3679 1458 1332 4297 1370 1917 3977 2856 1603 2132 4016 1291 1613 661 3260 1479 1202 4787 4464 3276 593 854 560 2292 4506 3659 4432 2664 1871 1646 276 3527 3173 4893 3200 2321 4203 3639 4801 2451 2683 1196 4228 3899 4391 3919 2917 638 621 3585 370 3800 1005 3440 4065 708 1835 4232 4887 237 2129 336 1627 860 3302 4168 4851 4619 4500 3178 925 426 3015 487 3112 4039 4986 1403 3460 2507 806 540 4903 3083 3989 694 338 44 3196 3403 763 43 200 2673 1397 1019 1874 1820 3029 1920 4965 4006 125 4948 2895 4540 4182 3296 2375 4379 449 1366 502 3146 2072 3988 1913 938 3147 492 3974 3212 4757 4417 572 1689 1984 532 800 2000 2309 4224 398 3755 3857 467 2260 1739 4317 1990 3253 2110 3125 4919 325 3909 2019 349 936 2187 4940 4595 1424 83 163 1665 991 1809 5 4975 2169 1636 2333 1963 549 290 561 2621 3454 2709 3229 1945 1413 2484 4942 3221 3834 1413 4417 3036 2707 3850 2381 18 2443 2681 879 3825 222 4920 756 3883 3244 433 3876 2609 2269 2866 661 4403 1228 4688 3462 3185 3367 854 1341 1450 3694 4784 2307 3354 3201 29 4118 2313 4639 749 327 4978 3060 2071 4286 305 2626 4709 2242 137 2356 2668 1029 1157 4975 2695 3160 5 4954 1731 4211 1596 512 4232 1559 1879 1115 3990 248 4384 1955 2343 3069 2404 3087 1236 4321 1249 4627 3027 3335 3957 4171 574 4450 579 958 2287 531 975 1967 1555 4118 4612 4000 2741 3647 691 3936 1461 2313 2023 743 3296 4263 2119 2143 89 1575 539 4894 560 1427 515 2097 3858 1226 1020 3386 106 1615 93 4525 1463 1384 497 1280 1605 106 69 713 97 1074 2837 2532 519 1367 1642 325 461 439 2119 4787 4726 855 2764 1542 2792 3110 3186 3339 567 820 632 1005 4518 2776 4373 2414 2666 303 2335 446 3946 1508 627 722 4236 2436 534 4289 3881 56 2820 2003 4556 3785 4087 1172 3503 3303 131 4304 1934 4850 2367 3574 176 4277 2014 1034 1612 1976 3543 1462 1752 573 473 1121 215 3357 3251 3946 1036 3252 1828 4950 1858 218 4910 673 2351 4219 2800 4409 2480 1546 3276 2121 894 3321 1549 4253 1369 156 182 3113 1956 1143 1415 1943 2001 620 1825 2465 78 4834 3445 1139 3985 754 2159 4061 2088 2492 4489 1424 1556 1042 1141 862 4983 1555 690 3287 1483 3243 1316 240 2693 840 1541 3279 1986 776 1488 4573 3320 522 2096 491 1600 3826 964 2537 698 3969 1381 522 4059 1308 4179 36 987 2611 37 3991 1505 4701 4208 1475 2955 3738 4594 391 336 3745 3976 487 2233 1324 4556 334 1447 696 2955 2164 2022 3177 1478 4826 963 3678 4878 2604 1610 3265 846 2397 2068 3594 4860 4241 4734 356 1136 3183 3301 4511 4760 1983 2985 1392 491 4867 3047 3230 1268 1191 132 4398 1738 3253 3254 1786 4462 853 1485 3349 1188 4179 2114 1818 774 1451 4183 429 2917 3433 4904 3122 3585 3556 1440 1862 1915 1234 4231 587 545 2194 3258 591 2634 974 763 398 2890 2560 4939 3071 2024 1878 3366 1333 959 4809 4381 679 738 3626 249 763 627 1798 391 3086 1501 684 1143 3062 1392 1540 2098 1700 830 1723 3867 1580 1628 4292 2693 159 1666 3753 853 235 2623 2042 2569 1505 3823 4820 2217 3680 870 1022 1253 2400 969 1873 472 1017 3155 3318 3203 1349 2573 4948 3485 763 874 1410 161 1021 1314 4394 3457 1685 2781 4670 3450 291 3211 3783 2535 3186 2497 1564 1211 284 2695 1400 4449 2643 4448 3175 3894 2914 4233 4202 1081 2907 1797 1993 2994 3514 2288 801 2297 1137 3883 1934 888 869 600 1923 2940 1193 2430 319 3303 2506 4711 2202 1386 3964 4856 16 114 1219 64 3213 4200 1876 373 3450 2693 3012 491 151 4573 184 2647 2402 3443 3519 36 1983 4014 2982 4627 1904 803 2979 1328 2390 2907 1604 466 3769 2503 4768 3907 348 3697 1061 3010 1445 3228 3036 4717 4260 3566 483 1762 2897 1068 1042 2920 2955 1351 3359 1249 3030 607 3133 3051 2485 3413 2251 3872 3153 861 4049 2490 3616 436 2760 2619 2129 2726 1859 1644 4583 331 1520 956 3500 1178 1612 1308 2400 1623 3270 960 2594 2609 314 2200 4878 615 3729 864 144 4685 1534 2661 3432 1897 1029 849 3215 4547 1431 305 2217 4236 1153 3157 3083 350 3158 4071 3054 2580 3722 1160 1588 2563 1616 4519 2138 2770 3390 1507 2744 934 608 3619 2621 2801 3826 1388 3438 2908 3458 4316 665 194 4533 28 1540 2622 3360 3507 60 1984 1322 3520 4259 1453 49 2568 3705 1981 4186 2369 2742 2424 1637 2454 3111 4483 3640 1124 2559 2256 3323 2948 840 1964 2411 275 2026 3428 3043 1494 3203 3674 2208 2913 3112 4421 3372 4026 4194 3128 2437 3754 2421 2204 4743 3604 2541 3287 4095 2913 1361 1199 625 2826 525 2778 1141 800 1483 2896 13 2265 4674 2035 1955 716 1674 3541 3560 2230 94 3867 4528 4165 2850 249 420 3862 4887 1638 3344 4418 3349 273 4182 1231 449 3020 4906 12 2880 3080 1296 708 3538 4762 131 2574 2266 580 3503 100 836 703 1905 1916 3896 3463 4454 676 4839 271 3441 3737 1891 4711 4257 1721 444 315 186 2822 2440 198 898 1294 4034 2863 4672 106 3231 1467 3608 676 4421 3451 4385 749 3677 2375 3539 3845 3418 4931 971 1004 306 3904 866 1816 517 598 4597 112 180 698 973 4153 2275 2891 1881 895 3402 1825 195 4886 2328 4167 4617 4217 1337 1048 4125 2755 2363 2976 887 4556 3475 4173 2169 854 2549 886 3587 1496 4371 1882 1728 1227 2317 1940 361 2332 938 3670 4336 3242 2374 4292 312 3536 4277 3866 3571 212 4743 1776 971 2001 2301 2069 3846 3289 827 970 4900 1400 2078 1824 979 2349 4551 3189 1541 4474 1720 2018 783 1911 3087 273 2549 2497 1457 3214 3989 841 3809 2547 1841 2450 409 2232 257 1467 4816 1722 2416 1910 852 1683 1830 3007 82 2587 222 4417 420 2742 1272 4414 2098 1870 4706 4790 92 635 3938 4922 2983 3743 1944 448 1286 2302 4978 2314 851 4558 2311 2155 342 2852 3832 2737 1349 4936 1268 761 3269 1945 3451 3855 1680 4388 1294 4445 655 4601 3428 2318 2025 3206 328 348 3373 4080 3569 1183 2677 1439 1362 4462 148 1418 2451 4499 1505 43 143 306 1988 2124 257 3994 4280 4210 1352 3159 177 512 1682 783 1408 1015 3336 4700 3901 4061 270 2473 909 404 1943 3201 893 2093 670 3604 37 3796 2376 4123 3784 1006 536 3710 1987 1032 2826 1900 1179 3465 2676 955 3476 3331 1218 2560 3896 2366 3538 1242 4108 2269 1495 1474 297 4821 154 2717 2146 2628 4093 4253 730 2175 2353 4237 1528 4558 419 908 4973 4947 763 2523 212 4625 981 2258 4867 2113 1043 432 4938 1163 3059 2818 1034 939 4549 4622 1600 1282 2024 4776 4849 4447 2989 4157 4808 3341 2739 3086 3894 3911 3349 4135 3241 4913 3537 2553 4888 48 4675 2036 1238 2293 2505 3345 1798 4246 3416 4991 3414 2814 2959 204 4190 799 1170 2711 2834 3708 690 3915 3534 1235 987 719 4362 2523 2094 2012 3055 379 536 2006 489 3643 771 2367 613 2167 3835 4080 1172 3057 1524 4321 1128 1670 2610 3723 4559 3068 776 2076 1046 1961 1213 626 16 1328 2693 4636 1183 2008 1180 4806 346 1510 2696 163 992 2892 1567 2800 3635 3922 709 2414 761 1124 4467 1136 807 1603 4566 2750 4070 825 780 1794 1283 3511 1972 1465 417 4697 3373 1627 4802 501 4879 4054 2470 2703 1586 1648 1971 2865 804 741 1974 689 3119 4552 2041 3348 4762 2092 1269 1721 2886 2813 4418 4569 4940 1738 2925 1206 333 2175 792 768 4522 97 608 3112 3085 4008 1988 3423 464 2060 2449 3822 1389 2617 1134 2340 3597 2757 2868 1011 4949 4092 2013 2772 3365 1737 1362 3453 2259 112 2661 776 4960 4778 4222 2557 1705 616 4132 4138 3409 636 354 907 4085 2404 1133 933 1672 4440 1811 4215 831 4463 2374 242 2920 3708 2176 2520 799 383 3857 2300 4057 4929 1455 3640 2301 96 1262 2405 3959 3932 2648 1486 4694 1105 1959 326 2417 4502 369 1011 1918 4573 674 1875 4339 868 72 1950 4049 1892 3003 4892 3045 2280 3172 3657 3944 3423 3640 3389 4266 1454 2316 2569 1458 2728 403 4495 4866 2343 1302 1313 4969 2960 1732 3109 3424 705 4082 1680 4857 4822 3900 4184 3220 597 4028 4062 4924 1695 3402 3340 4314 1440 598 323 3528 1936 3970 2425 345 1099 3395 889 4263 3278 537 466 1573 4443 957 1877 4751 2783 1683 4121 4246 2386 4704 1009 2208 3358 2861 1281 1919 331 3061 2845 4808 3515 4268 251 3552 4874 4224 3505 2762 2072 3956 1010 1324 4810 3474 3565 1707 3965 4433 1422 851 1653 4671 345 284 4749 3969 4549 497 1607 2993 157 2883 4599 312 813 2013 2579 2388 1555 2641 2702 4027 2883 4204 501 3200 3000 685 4449 4869 3888 595 3374 1122 4311 4552 2324 549 1558 1296 4716 1939 1982 87 2445 2990 1335 3322 3023 1976 430 828 4183 1894 1471 644 1717 496 1135 861 435 352 4861 3871 4031 577 4842 4628 2122 1602 3502 3397 610 3539 2197 1901 1660 1671 1818 1206 4822 3635 3611 2395 2885 1646 1541 1142 162 3442 2545 2859 3243 650 999 4437 3839 579 802 2112 3609 1304 4885 3710 1208 4152 4380 1861 278 3755 38 3921 3096 4038 2576 4934 294 1475 102 4025 740 2213 1502 3752 1008 2402 1734 3350 2285 3625 751 4516 3063 2398 1289 1980 471 3931 4034 1112 2609 658 2617 859 629 344 200 1949 787 284 241 3221 3163 4169 2511 362 3059 192 344 35 4290 2074 2410 4640 2421 1108 3308 1320 2791 3516 3055 1084 1927 4594 1725 940 548 4043 4584 3928 3324 3314 1126 490 342 3767 931 427 2772 477 2887 1746 1814 1007 3194 3428 435 2763 2614 2136 2368 642 1340 4123 2782 1262 73 4220 3647 1572 4690 3344 4753 980 4701 424 1158 3719 2354 1433 4715 4353 3731 3037 3320 2063 4680 1576 1512 99 4189 2670 1066 2997 2408 2323 3150 4924 918 3587 4988 4237 1941 3579 4273 1107 687 447 2068 4992 997 4550 2504 323 554 4595 2706 1835 3138 3461 182 3259 1331 65 3324 4830 4475 4230 1129 3324 826 1993 3372 2758 225 2770 1206 3391 419 815 1653 2347 1843 160 411 4300 3814 2171 3674 1779 4002 2673 1091 2028 2344 456 4016 3645 2686 4298 1630 2776 215 222 1335 486 4287 2734 2758 4935 778 153 1612 1032 2964 3401 59 1836 2815 4351 3129 1070 168 2558 704 829 4378 1653 4516 3505 1880 2273 756 3069 4509 2427 3452 726 1870 37 3490 3019 882 1611 768 3390 433 2552 3193 354 2403 3086 4479 4908 922 4602 3909 1495 131 3223 2705 1195 3289 1578 892 2566 4674 54 1964 2805 4878 2161 314 4029 4237 2246 3288 1717 1177 645 4445 2990 2436 4856 4868 2286 250 2815 2945 497 610 3638 2966 58 3377 1549 614 3104 528 689 3920 380 1608 3569 797 4522 2123 3643 3773 2102 2892 2696 263 1985 4385 4509 712 3220 1832 2115 174 4484 2368 898 2885 1695 1 4578 3082 94 1895 1442 4566 2059 2819 708 2093 701 3594 752 559 4425 4130 1865 519 4219 2499 261 1626 982 738 4944 306 4208 2500 1371 2084 1497 2944 1201 1575 4654 213 563 3422 3545 1199 3708 3704 1171 1556 3591 4022 1821 908 1064 4668 3979 1568 4856 1499 4866 2274 1885 406 3364 1414 2025 1156 881 1687 1170 1726 2008 1396 1362 978 3263 4133 4144 3559 1415 1689 3311 2737 1219 1169 2421 2287 730 2467 1638 187 731 367 1042 962 2521 598 4356 4429 3983 2637 3453 4823 2557 3508 4985 4225 3428 2627 3344 4170 4855 751 2823 1786 2802 4786 2280 840 807 1609 87 1818 2151 1410 1431 2046 1886 4317 3765 2735 4802 4130 4002 3970 4109 4083 407 3890 2281 2924 1316 400 1787 1048 3945 2751 2779 3558 2897 2375 187 3724 4212 1488 4262 954 1820 1082 4612 146 2946 642 4523 1430 217 3463 807 1739 2203 4974 1317 421 52 3 1612 2659 3020 3185 1150 3445 4048 1828 22 1315 3619 3028 457 2015 4657 613 186 3973 196 4221 3657 1730 947 3878 3044 1171 2783 2909 3593 216 2992 3617 4978 4513 296 2680 3402 2871 4190 2022 3335 734 3517 2312 152 1233 4374 3840 3273 2231 1644 513 3741 1381 4254 1446 2011 2620 2386 2934 420 3796 24 931 1100 3313 2781 1259 1583 2189 993 2599 3326 3342 4179 642 1554 481 4083 3547 3713 4663 3480 831 4566 4735 2420 3056 4613 1333 1852 3491 3599 113 788 2825 1603 4075 1387 452 998 374 4115 1621 2301 4346 4243 1363 3896 790 1648 2638 449 1557 3034 2452 4181 1465 1939 4644 4626 1942 2085 4522 2026 48 3068 4763 1054 2895 1554 1593 2975 2170 1354 1201 3273 1730 2415 1018 4130 121 1382 4137 2601 1514 2160 1931 4426 2546 757 570 3233 1319 4590 4708 4548 3640 2169 4560 3126 878 4858 915 2634 3178 800 4642 3365 1791 230 425 1303 1128 3507 1187 1698 4213 1620 1251 3700 1506 3800 2010 1258 2417 3701 4233 4342 626 2609 690 3899 207 171 4458 2813 3693 4219 2066 2102 4136 4823 4134 4951 93 1304 2053 604 1574 3933 4971 355 410 1910 795 1934 1499 1587 2017 2303 2668 2054 1207 4628 4147 4734 1998 3451 4243 4829 1479 887 3917 3396 434 635 1296 662 2351 3430 4902 2755 3084 4841 2713 1214 3879 3133 374 1016 3483 3656 592 686 4288 2905 3200 3081 1510 2202 1212 2829 3242 1830 2366 1516 1349 2171 407 2755 2817 89 2740 4573 2086 2555 3263 447 2796 134 4160 4821 25 1880 2686 2907 3765 4803 4488 885 244 514 3449 2099 603 3379 724 4815 3175 4934 28 2441 4843 1205 2480 941 2209 460 2141 4171 1971 2858 755 1432 4537 944 3628 4027 285 2730 675 3883 4718 860 1730 1271 1930 1928 1310 4525 3176 325 223 1023 1636 2041 4290 1778 2956 2074 3585 2321 947 865 1903 2115 1420 1376 564 4112 2386 1501 4431 2873 1885 1736 1525 2656 175 738 293 4174 2351 2463 2446 1788 4294 1965 1651 4300 4800 869 1534 2353 1622 1657 2122 1500 1042 4727 2565 1905 1490 3877 46 2025 736 2661 4700 2785 3945 2715 544 1194 973 2565 2851 10 4031 1058 1354 4092 4718 1878 3545 4867 3778 1309 2014 981 274 806 2328 902 838 1186 1816 569 3465 1412 582 167 3665 3707 4855 313 2514 1798 1749 1336 2184 3504 3504 1800 652 3925 180 215 4552 4371 2187 3620 2494 3212 3316 510 2094 3235 2883 571 4302 92 2659 4227 952 898 2722 1323 4718 659 1917 4470 2441 894 979 3673 4818 1321 3390 3502 3968 1732 2095 4243 2580 3802 4849 2659 1505 3936 1401 230 681 2990 4116 1909 1467 4415 258 2385 4065 2900 2522 4028 3609 4997 2742 3791 3170 4402 2055 206 3866 4991 4508 1577 3077 3223 3712 2420 2673 4610 4369 2274 4304 1490 4528 3837 3765 4026 4118 4489 770 528 2964 3842 4618 1685 3294 1835 2626 66 2520 560 4510 4317 2201 3878 4641 2612 3615 4617 388 1217 3796 2263 611 231 1472 2084 2053 1372 1502 4228 4323 2112 1267 1960 50 2005 453 2547 2148 694 4752 1781 3515 3357 4031 2770 56 4664 200 523 2955 4892 3305 4661 1012 4877 3378 704 4586 1037 155 491 2410 4475 152 2878 3538 4413 346 1273 1126 2465 1183 4763 3082 2096 4933 477 1913 3100 3267 1160 4870 3108 3760 692 3417 1582 1144 202 3045 4476 1668 533 4224 4930 2034 3076 4370 1470 1877 651 3225 4809 3942 4619 3835 693 2083 1519 4310 2128 79 2929 2981 4584 2105 2716 2780 823 2594 3067 1462 18 2641 1673 460 116 1258 483 1953 3680 2263 4995 1380 950 1548 2910 3214 1717 2041 1016 1773 2515 2060 2518 4255 1073 4133 3157 1715 3571 246 466 1221 1702 2987 4601 552 2162 4094 2241 3049 3418 2567 375 3961 137 4985 947 1547 4608 1376 2456 3103 3286 4794 296 2478 3343 107 2271 4879 4583 3014 2782 3825 2673 1465 2299 2666 1987 3208 1943 4536 3237 1093 3465 1374 4874 1630 3721 2049 3981 2199 3855 1371 4488 1330 3077 392 2961 4676 914 663 4646 4506 2385 3667 714 4075 3889 1475 1536 1496 166 1183 4792 4993 1536 2764 1456 1322 2674 4753 4093 2702 1598 619 2194 4411 2431 2022 4439 2494 2775 3429 30 792 322 2492 1554 3434 3018 3155 1009 1012 3397 110 452 1964 2460 475 1668 3061 3997 3115 1071 4390 435 3085 1267 2637 836 2048 4698 3764 2140 4575 3567 3802 4625 2162 25 2165 1702 2082 2165 742 3411 788 1152 3013 1405 1659 4389 2776 2562 2407 1220 1542 4648 3586 2060 2609 2758 3037 2530 2531 2827 3702 4114 3463 2541 4661 3366 1204 2557 4672 1717 1598 130 579 1309 1925 1907 3899 684 3858 2457 1325 483 1750 1883 2370 1485 2318 4644 4375 1902 3389 2554 4682 1015 1096 0 1559 1225 2567 495 540 1240 1418 3225 4193 1675 3563 334 1821 3713 3323 4218 2613 4981 3687 2622 3421 1941 4452 1621 3868 887 936 4107 1819 709 2716 2681 614 660 1583 3796 807 2263 4972 1047 1463 945 1103 3118 3247 4355 651 901 2438 2170 3800 2407 531 4620 3627 267 1141 3269 1391 3100 4761 1821 1267 2177 868 596 819 3572 865 2874 4491 3934 2877 1920 497 59 274 50 19 938 4321 217 1849 2920 155 4508 1388 1909 3706 4167 1511 215 3051 2303 3853 4830 2551 2908 4460 3768 2074 4416 832 3669 2971 2575 3778 3253 3734 4263 1994 3049 1902 1394 907 368 791 2074 1206 4323 1962 3877 3066 2267 3274 3361 43 3433 2712 3034 2232 3390 4530 2025 4546 1252 4705 3295 3521 933 3194 946 1841 4098 637 919 4589 4327 4669 1585 3482 3965 895 2872 1672 600 750 4998 796 2226 1286 2662 1641 4815 4832 1326 1417 1899 622 1335 614 1393 1960 610 1750 3470 4941 341 2591 4972 4989 3327 3311 2043 607 2960 3487 461 4535 2700 4536 2433 563 4281 4039 484 3317 2271 3085 2274 1155 1360 3722 226 2606 2024 843 1459 3280 1927 4618 4777 4225 844 2181 3445 1859 1660 217 3860 2421 1284 1080 1727 3485 3859 3552 1203 4058 299 3655 3257 1856 3209 3124 1929 909 2211 2936 346 2004 2043 1988 1813 3584 2009 4044 43 2227 2829 2413 3940 1471 517 475 1994 2581 1467 4854 78 396 1875 1926 2713 2861 3060 3573 524 1917 292 1003 4396 4818 4465 496 2520 241 1878 4781 2926 4943 3802 2868 1678 4718 2263 1393 1572 2731 4586 1465 811 3878 4486 3090 4416 2194 566 4203 1173 2181 4803 4950 3783 1082 3146 3083 635 645 2602 1663 1860 590 2716 3220 3112 1698 1108 2739 3011 2239 3103 1911 2708 3087 1645 2036 1895 766 3953 1392 1331 3687 4143 2145 669 4739 1142 4935 492 4980 2358 4963 830 2106 1187 2887 4883 147 426 2040 849 3456 1547 802 136 2038 2888 3347 127 4074 4389 3131 4053 3225 4433 3578 4197 4855 4452 4547 4260 2166 4273 3575 3573 313 2702 133 4231 2609 4617 4288 533 2160 4085 3289 830 4556 1542 3536 3302 2734 2681 2421 229 3556 3352 2707 2190 3380 3181 1495 4707 2090 3572 3325 4735 3388 4853 997 3460 2060 1874 4172 31 3709 2294 605 2592 4897 4191 505 2062 3018 4814 2805 4114 1176 4069 2683 1690 1851 376 3132 2434 1682 3734 4733 950 163 3713 4295 2707 1283 876 3 990 1352 2390 3989 517 1589 3445 4624 1261 3014 340 2211 4492 1317 1311 283 3952 4692 3190 121 3103 3836 1496 3013 297 1285 4458 459 2706 3495 4097 3093 1508 2921 764 3529 1373 2431 2933 420 4973 1465 58 271 1283 3482 2527 282 1680 1447 3950 4206 2980 185 2385 2543 2839 292 4135 2698 4795 1184 2798 610 3896 4671 681 4858 1456 657 4546 2654 1485 4469 1104 3234 4471 377 2260 1080 2330 1779 4625 2140 1695 1097 2419 2394 90 838 2737 1995 2604 2806 965 2723 218 4676 44 1382 2662 3769 1729 4173 532 472 255 659 1872 1744 2775 3541 4270 2706 4590 4522 642 3979 1939 263 9 1700 4896 3182 903 3856 1910 991 3450 3894 1934 359 1951 776 1189 3593 3836 3043 4999 4347 4134 3296 128 4922 620 4130 2863 2496 2884 4441 1779 1845 2540 4006 4883 1635 3987 264 4433 1135 3553 638 4070 4589 2420 1654 652 387 4028 4174 1391 481 3042 1713 3630 2710 2644 2772 3097 2751 1012 4057 1563 1996 4075 4539 1610 507 2288 2764 3893 4106 1933 1108 3813 1879 4459 2883 1102 2026 3492 2136 3968 253 2138 2946 1386 3018 4663 1543 1358 2518 3220 4212 1403 4732 2454 4133 3630 4566 1301 4757 1138 4480 4215 1624 3162 1653 4061 2446 85 2374 3828 3627 3602 3167 3335 4453 291 2188 4139 173 1118 109 3299 4490 737 3766 269 2042 2439 3050 4356 4451 809 2065 239 3089 2062 1111 51 1528 630 4455 3084 461 3571 2040 1914 1888 4962 191 2753 3508 488 4380 2873 979 1728 1969 1024 2643 1404 4585 2612 1021 802 3062 4062 2064 4025 766 1315 1121 4563 3758 2989 1271 102 231 449 2948 4342 2224 1938 4286 4453 2500 862 734 326 4678 2975 4896 766 2802 597 2381 2804 3366 3954 4693 3810 1673 2879 2649 4732 2698 2789 3113 2985 4412 2118 1297 2864 2649 1268 361 941 2180 90 2852 1276 1388 3058 2881 3708 4646 2903 177 4528 801 4101 2776 1978 3200 4829 1808 2241 83 2645 2626 3261 1800 2927 1566 983 2753 609 1942 4751 4445 677 1861 345 755 194 3565 287 371 1954 3098 1941 120 1205 4929 2568 796 4551 3030 4753 2722 2469 4340 2769 3458 4486 429 717 2984 2967 3820 3091 862 575 4865 3789 3899 1981 4038 199 3547 3007 264 2572 770 652 3192 957 2148 3332 2250 2652 3278 3086 2365 1830 2690 698 524 1799 3486 3872 4749 2811 1311 3337 3864 1658 1101 773 970 1083 518 4148 67 3900 823 4809 1420 465 1560 1987 1373 651 4648 70 4238 2099 1648 1602 460 527 1868 1995 1991 3164 3984 2577 2898 2789 2336 2600 1690 1915 3707 4602 2402 502 1352 4724 4126 1283 3880 2470 3807 1718 1118 3215 4311 3177 261 2304 3653 1989 4362 3528 4003 3049 2618 3048 278 4281 1740 480 3110 4616 3850 354 366 210 2051 3428 406 1011 1733 1260 4464 4308 1674 455 2056 3767 2305 2971 1183 1224 4662 1111 1057 3767 4260 4031 2517 1484 378 1107 2797 2978 1691 1290 1328 1343 619 403 2244 1890 3822 3741 4223 3438 2227 2784 4832 3286 4330 4907 3937 369 626 3295 1849 1265 3734 3172 2095 1310 1777 931 137 968 2359 1828 4158 3781 2837 4344 1744 1336 1628 2057 470 147 1154 4337 2888 4718 666 1712 1005 2048 4805 3922 1890 3357 2481 953 4068 3928 857 2311 168 1659 2685 2518 928 3430 4723 931 1129 868 3246 24 2091 3732 2686 3466 520 4526 2256 2586 2977 4908 1680 2744 4581 1339 1948 684 1156 530 2234 2359 283 1603 1653 3978 3520 4447 1988 1756 4976 4951 4012 450 1711 2280 1287 3274 3216 511 52 4562 1950 3621 4889 3727 3443 2308 4127 1279 350 563 2453 4634 762 383 488 1887 1476 2019 4505 395 213 2152 342 2816 690 2136 1118 2779 512 2880 1087 2502 3756 4420 2771 2824 4345 4072 4859 1177 2379 36 2217 1914 1468 1599 4260 292 4149 2352 3536 2409 4424 3984 1965 1128 4123 668 4961 748 3136 2554 168 1871 194 3489 3355 1899 1050 4078 2758 3258 2243 444 2463 654 155 2420 284 3843 4091 446 3347 2036 2545 1063 1384 2763 844 1803 3067 2397 3631 1252 952 3197 106 4107 1067 336 2437 1647 4755 1626 3142 4844 2750 4294 4693 2422 3021 3947 358 1790 1826 501 1863 1948 4951 2394 1723 4422 2609 1623 4662 1407 4869 377 530 43 2794 1200 4201 1274 3319 4824 4780 1780 2334 1027 307 1892 3641 1452 1278 1200 4323 477 324 4303 1795 3595 833 4500 2693 2072 1781 90 1665 2369 11 3321 1816 1769 3231 4328 2004 390 3364 819 2935 4419 109 1919 4927 1198 610 3443 2585 1082 4918 969 4355 4472 3214 3502 4350 1073 124 1246 3860 3199 3597 2524 1646 3005 463 121 2137 1996 1608 3089 4898 3327 1795 2070 208 1929 4156 384 3585 3422 3830 4426 615 959 1482 3915 540 706 3251 1763 588 794 72 4492 4419 1154 257 4300 3359 3912 422 2976 3798 4751 948 3786 1122 2453 1450 4374 692 47 4827 4562 1872 130 3148 1484 1152 3964 3769 400 3227 4841 4187 2090 2503 2842 4325 3225 1388 2707 4991 349 3835 980 1042 3147 4320 3874 2997 3254 4203 2727 799 903 501 1352 3302 4048 233 3589 2711 558 4768 3398 503 4206 1891 2212 1194 4593 2490 549 4234 4410 4442 2444 131 872 3063 1201 3572 1959 4110 2445 4216 4852 3886 4592 2462 3864 4831 1006 693 276 444 2888 1244 751 3806 4855 3354 4044 2937 206 809 4847 4210 3793 1392 2857 3393 3968 1689 914 2331 1632 3743 3519 3260 2652 2653 3052 2851 1856 1219 539 4334 4263 2906 2662 3632 2277 4571 1857 616 2279 1642 140 891 2193 3657 1871 4661 2848 4058 1031 2716 4706 2092 352 4862 3931 171 4219 1892 1530 2377 1435 3830 1843 1257 1898 913 1269 4270 2953 1191 198 4631 4792 1934 726 4805 858 3995 810 2200 317 3272 1994 4863 1744 1548 4125 2279 998 4250 3563 679 3227 1333 2780 265 3676 4890 4535 4762 4659 870 2461 3638 1563 2250 3919 3910 1865 2372 1420 835 2268 2240 1344 544 3191 3414 66 2747 276 1918 1245 1684 3051 857 2058 3169 2539 4498 3995 177 1176 2478 930 1799 1470 375 1051 4090 421 3543 3848 3955 870 3715 2052 2742 3132 1637 882 3541 502 511 2079 2145 3220 3661 261 4178 2984 2834 3553 3065 1880 3310 2535 3756 2625 4542 84 4658 86 2340 2325 984 4432 4793 1558 4114 1880 1240 1301 4206 612 3109 17 94 686 4846 3037 4004 1067 2153 793 2795 3554 4510 2571 115 3091 2678 961 1206 3435 1372 3914 3921 4525 1194 3126 1277 2154 4488 4420 4182 1101 257 459 2523 2040 2830 1634 4863 1635 4943 283 1623 2087 2766 3629 2990 2493 1871 74 3285 1612 3790 4049 199 971 1103 2629 4892 3547 4723 629 484 721 3353 789 471 3515 1036 132 4939 2257 84 807 4693 1310 847 1671 3227 3710 3905 3031 1729 3706 4362 3225 3754 2386 4631 1225 4050 3595 3603 385 562 3713 4032 2071 636 1518 1736 3212 3919 4828 1153 2062 1733 1928 1528 4952 411 3415 2493 1888 3560 2909 2864 4770 26 3411 1050 987 4654 1705 3328 817 3902 1413 3083 2889 4296 2098 2747 2848 228 4501 885 4030 358 2052 1434 2864 3397 3768 3274 685 2619 2588 4617 4844 959 4710 4329 3204 495 1926 4408 275 425 2876 996 2739 620 3175 2428 4386 1562 1915 3944 4841 537 4435 4964 4863 4243 3488 3817 2142 2866 1288 3788 2975 1696 202 363 1146 486 3481 2901 4321 4859 3183 4650 2565 1664 2844 2451 3768 3817 2039 227 4605 126 2004 2598 469 1507 4551 848 964 3917 3888 4554 3755 1654 4832 1977 1767 3659 2466 943 4737 132 3973 3925 3063 1299 1610 2801 4236 2161 329 485 420 1565 1792 2592 2570 790 2726 794 1069 2739 981 2170 1840 3985 103 1153 4537 4166 4619 676 4445 3443 4607 1790 1378 4055 4080 1237 2396 2729 1032 702 292 718 1007 2051 4118 356 2037 2540 2826 3118 4495 2972 254 3345 3008 2341 4306 3510 1386 862 4732 194 1541 1980 3868 4578 2807 4367 4922 930 1652 1023 4751 4728 4264 2550 3666 3277 4763 832 3052 28 2069 536 4854 358 1562 4148 3441 549 2783 268 1413 55 1182 1855 2807 2101 4982 3808 1853 373 1255 1287 236 4700 820 586 2596 824 2513 1181 3313 262 2679 55 4326 4919 342 2111 3525 2561 1771 3743 3674 1435 1498 704 3521 3744 1902 22 2757 1703 3854 2726 171 3554 1802 4357 125 3543 1847 2187 1091 1268 3246 1728 2423 126 3876 1461 2056 3643 950 3910 762 4246 3715 278 3948 1747 3342 1252 1106 1539 564 4037 2195 1562 747 913 1070 176 3652 450 3617 77 330 4603 4034 1113 1555 1590 1281 778 327 4473 293 4367 2846 1489 2310 3469 2647 1815 4900 232 2983 3074 2630 2420 1902 614 3077 4276 4840 3963 1802 1611 130 505 3573 1753 2724 1587 2664 1072 1782 4681 4947 1952 2702 4946 2745 1391 3337 1041 3275 4637 1503 160 2547 4408 316 4362 1092 4827 44 414 3544 372 3487 4569 1594 1350 518 4698 813 3980 4605 2092 236 351 2041 4880 4140 2577 1875 2105 2663 347 4773 4236 1698 389 1749 3607 1193 84 992 4417 3186 4289 3649 2475 943 4015 1866 3870 1160 4077 3257 4812 2272 2725 3951 1291 1295 1979 3294 1496 2864 4458 4697 4480 1819 4420 3832 3109 2294 2041 2131 2518 1520 3106 4039 984 2838 2598 1951 3972 3977 3828 1309 4257 0 4898 313 36 1530 2437 3337 708 974 34 274 4227 3327 4605 4373 3706 602 3656 3934 3188 4090 4454 4417 4631 3558 3145 4750 1806 2593 281 271 154 194 4733 726 4433 3077 4143 4621 87 2458 4411 2688 973 3620 4704 3054 1248 3461 2698 1200 1828 120 1087 256 4032 2215 3745 4726 1099 1246 1139 759 543 2463 1600 1352 4733 3384 4640 2169 521 3378 892 4685 4679 2083 828 4192 430 3279 4599 914 3533 282 875 1132 347 1289 686 672 2507 1161 924 1334 3472 2016 297 3836 341 4651 1593 646 672 4327 1518 566 4464 1245 1979 54 957 2528 1831 2691 4791 0 4043 4028 281 3324 2295 1332 1383 2243 346 4660 838 4123 433 4157 363 4990 3322 4355 286 2034 369 126 1376 1152 1739 2006 4148 2048 601 564 4554 3378 935 1105 1617 928 102 2545 4470 1378 1398 808 2312 2426 2584 3776 3598 2265 4963 9 187 1057 4725 3643 3392 3302 4349 3824 1331 511 3086 3429 935 1504 4783 4162 4916 2845 2137 2872 976 1904 2264 2795 4182 627 3294 2416 238 253 4540 4496 3437 1022 602 3493 3183 637 865 4461 2146 1653 735 133 4509 1379 3402 1336 1817 115 705 4664 4270 1907 2944 2740 2209 264 1146 4123 2940 2480 2964 3807 4718 2152 3287 2541 349 319 2998 1973 4546 2578 3729 1085 2244 2521 266 2011 1736 2146 774 4066 183 389 1325 2404 1000 704 158 2897 751 3923 2964 1343 1486 2503 4038 1907 883 3722 2582 2359 4794 2686 991 484 3037 900 661 1906 169 2874 2656 2709 939 564 417 3722 3475 4682 4152 4835 2951 3775 692 3447 1739 730 3381 4270 3776 3672 3799 723 1930 4745 176 251 3247 3394 3300 1381 925 140 2094 3786 691 2634 3346 1039 330 0 2385 3753 2445 2674 4652 32 2327 2787 45 567 3876 4976 954 2347 3843 4356 443 3865 626 2173 1479 3741 2819 4078 3569 977 1054 3246 593 35 3342 4521 1152 26 4600 478 164 1722 2337 2993 2029 4989 1264 3890 703 4917 1196 4949 1921 831 391 2547 3613 4162 3515 3193 834 4521 36 3264 2758 2637 4830 2245 487 4519 1712 4026 3740 3780 242 1268 1071 521 179 3118 358 225 198 22 1778 3909 1917 2350 472 2459 3990 518 3606 147 3109 0 560 1574 427 2236 852 4655 238 1069 1475 3342 3985 2596 409 3939 4021 4580 4046 776 3135 4205 382 4650 1508 4789 1246 3625 227 2988 880 2374 4906 518 2609 2517 1308 854 289 3347 1329 1881 1818 4953 3126 4793 4252 2506 1422 4948 4321 409 2841 1660 4605 2863 2397 3896 1944 4524 2058 4834 2615 1747 1986 2258 4863 3547 1961 3363 36 2298 833 4877 3337 3333 1 3507 544 2278 1871 4038 3317 909 459 2682 2822 2147 2511 1310 3694 1659 2092 3122 2506 2065 70 716 2867 2383 2418 4556 69 2005 3222 3761 256 2305 259 4987 2037 2568 2855 1418 523 932 2678 4862 4378 4359 1706 1193 2743 2246 4016 3283 754 3689 2876 2300 2821 1386 3792 2591 1870 1220 969 381 1382 3638 91 1079 1379 2016 1023 872 1083 4309 1729 1394 121 2378 1230 1384 997 4302 4294 813 4063 234 4982 2635 1045 2555 2202 1229 3422 4182 2937 3569 2877 3203 4537 3696 1025 1445 4413 4865 4724 3977 1460 3837 2403 645 332 2243 4406 2091 470 1271 105 805 3428 3791 1210 3067 384 1527 3676 4460 2151 4141 4323 2049 878 2583 4400 3776 1299 2843 4063 3047 2424 1047 80 3551 4115 1739 115 3444 3401 1355 824 3834 2181 2593 4877 1970 1083 139 3089 3877 4098 1593 4353 3632 241 314 1432 184 1345 3471 1837 2040 4555 4712 2440 4817 148 894 4316 1906 3950 56 1363 3240 4369 1778 422 3487 354 4323 3621 282 3426 2405 23 511 1530 2835 3553 102 1141 1862 364 1116 3336 2120 810 1955 3461 4413 1625 124 1830 4734 3908 1736 584 3832 1877 3715 333 1397 3200 4778 2247 1738 2972 959 2498 1152 395 1754 2513 4574 389 74 2020 3927 3023 2428 2176 680 3473 387 4528 46 3841 3057 2811 2693 1162 1157 933 2051 1257 4494 4066 1734 4962 4583 3672 1875 861 1619 909 2921 1703 1122 2177 317 111 261 2949 2832 4718 695 3380 1782 3586 1428 496 976 2013 1726 377 4268 961 3117 4314 3754 639 3935 1124 3907 2593 4970 4239 4126 2524 3341 3686 2611 2378 3805 3290 2683 4798 3151 4730 263 1001 400 4482 1048 3821 3205 2969 2167 4753 2593 3119 607 2004 673 2007 4520 4733 334 1809 165 650 1523 4739 1655 1631 829 409 1803 506 1731 1003 1945 981 4014 3070 3959 3331 3555 3022 1630 3271 351 306 4316 1177 2163 1248 1585 162 2800 2618 1693 1770 4027 1841 2202 112 3964 3978 438 1522 1124 166 581 4389 102 1494 3276 2407 376 926 208 1427 3153 2090 1120 2630 2109 4855 3040 2727 2407 3131 3584 3924 4926 4271 3533 4867 3563 2291 286 2645 2354 3499 3289 3594 1983 4145 2943 3033 907 3156 2749 2871 479 4383 3709 1490 4392 4380 1884 550 4060 1203 461 3040 164 4621 1062 1145 1538 1110 26 1029 2156 2827 23 4700 2306 2344 1656 1777 2565 1189 1397 2217 1136 2794 1218 4702 2532 4688 21 485 4846 1618 92 2397 324 3796 2503 3466 288 891 2553 4480 835 1053 797 315 1438 2106 2863 1398 1869 2587 4968 586 4421 2090 302 4866 224 1186 1116 3126 3950 3856 2765 4860 3398 1011 3935 85 1005 712 2213 2623 4943 1089 3997 4014 1308 1375 388 3705 3284 453 1807 1139 654 3967 763 1934 3036 2595 4967 2799 1044 4326 2448 4486 4787 1026 3733 228 3675 4887 2565 2970 4599 4696 302 879 4867 23 3343 626 377 3972 320 4697 3946 2585 2152 978 1178 945 1439 336 4055 3595 2325 2145 2299 4682 875 2389 3767 1887 3617 3152 99 1945 3759 260 556 2083 1752 281 1893 102 588 1895 4433 2961 2701 2190 32 1047 996 3075 240 2464 2029 3829 77 58 472 3265 548 2113 2361 923 1380 148 1250 1942 3375 4050 4856 4302 3984 1349 1683 1563 4972 878 2626 881 1854 3786 4167 4318 2342 2587 3174 652 746 1850 4396 2128 4304 659 2237 4153 3554 3233 4718 517 1579 1581 1645 512 4673 1833 1582 1271 3283 745 4127 4716 4077 3364 4846 1107 1362 3623 1593 1612 3265 298 655 377 1347 4016 901 299 3604 3441 1707 1008 4448 2258 2024 1321 2160 144 555 3393 2037 1665 2548 1479 1004 1756 1459 4752 4859 3849 1799 3062 2307 1187 2321 4758 1246 1133 4689 241 3155 251 3129 2713 2287 595 114 2023 3750 4260 4476 2970 2699 1902 754 1842 4348 2018 4100 301 3650 1243 2518 3706 1111 844 3066 2445 129 124 2458 1675 1774 1992 1656 1430 4192 3568 2553 267 2459 2165 2047 2081 524 4952 287 3829 3229 2751 2395 3772 4574 1244 2232 136 862 4446 1626 4887 1843 1564 3974 2952 3134 3071 1632 3928 2514 4007 1359 3820 2638 2385 2574 3776 1257 168 4676 583 1406 3098 1030 452 497 2916 3100 1296 2920 1640 105 4500 4565 2421 990 2010 3315 2904 1930 1247 94 2746 3299 572 3816 4530 3192 2605 4970 2133 1130 2601 48 3283 3599 1596 3072 555 961 4017 2748 1994 671 204 4726 1592 1707 4671 3036 4219 2020 4859 4636 4776 2124 761 697 1519 3333 4625 1686 127 1607 3267 4426 4074 989 4965 1677 4850 4177 2485 451 4012 3325 2423 4247 436 3539 4112 961 3057 4207 3811 1027 2018 268 284 4320 1817 1332 4092 3525 4287 3981 1085 3293 4016 1920 4823 2784 1185 2555 1002 4076 583 3665 2548 4375 4473 3120 1397 4859 1537 4164 4817 715 2783 4378 1853 1437 1663 2823 2302 3097 2748 874 4143 3969 3268 712 3634 584 1870 1815 3993 3544 138 2356 3476 1847 3384 1297 4941 4906 1787 2089 4395 3941 3383 2550 1604 673 3913 1269 1519 3495 2270 1360 1185 1717 4231 4517 901 3062 3379 2429 3233 1182 3885 1199 2375 2191 1544 365 514 917 4784 2088 4498 1433 3755 4864 4721 2524 1431 2966 510 4914 2614 958 3306 1441 384 551 4866 4226 2415 1853 1156 1191 1345 1520 1845 4190 793 4070 2306 2885 4279 23 4117 478 1297 616 2579 429 394 346 4012 264 2163 2171 1723 454 4055 2166 1967 1339 1539 687 2008 2947 885 249 3758 2652 4364 907 1326 2674 4568 3432 3214 1085 4804 2782 604 1933 289 2602 1033 4017 1229 1185 4 172 2044 10 685 993 4077 4819 694 465 345 449 4257 3947 4647 3052 934 2323 1564 1609 3251 2589 3297 4090 2642 2523 1149 60 1549 1134 2588 3935 4605 2722 2768 331 467 1965 1448 3281 4470 170 4718 1944 2687 4392 1501 1897 681 2976 650 4988 1662 2583 4426 3180 2125 4560 4085 3649 2040 367 409 37 4333 4337 186 2210 2267 127 3065 4857 2944 357 699 2925 4896 174 4475 282 1401 954 1594 1296 149 1915 3302 677 375 44 2985 3825 489 1491 4091 3323 3118 3656 1034 2557 771 2374 1970 2011 1380 2267 2136 4954 468 2427 1747 799 2348 2753 3637 3861 1920 2992 4197 4187 3295 1150 49 2181 3618 2538 3891 1841 2774 1728 897 1117 4405 3323 4202 1960 2393 3475 117 1886 3259 1079 1724 1952 131 3972 401 2147 2849 942 1616 4999 2444 1926 790 371 1345 3102 832 88 4702 1473 366 3235 327 4150 3889 598 416 2270 4043 409 324 2083 1974 1997 1478 388 3289 4610 4151 2454 3150 1386 1735 1598 1799 2885 3399 1028 4026 2310 3190 1522 3882 4853 2738 4286 928 3039 3841 1043 4393 579 2993 2951 4226 638 469 1880 3706 1378 2473 240 2159 298 4664 1259 4140 2577 475 4655 585 900 4877 4076 1427 1445 1269 2491 1606 3609 108 4595 4424 2516 1328 3914 2671 1260 4071 4041 2446 2107 3200 3035 2860 673 919 1648 1030 1833 1471 1645 4014 770 2061 2046 763 154 4826 4710 1371 3086 4545 2902 1238 1941 4130 3996 2894 1997 1606 4334 2540 2589 3756 2832 2468 2894 1467 4211 691 1940 89 2918 2512 3281 1084 4680 729 2115 1047 1215 368 4229 2599 43 1950 4636 3934 1015 709 4667 280 2097 868 377 1996 2647 4591 2096 1238 3964 4126 1482 1002 2651 1207 3471 4483 4708 710 2370 1309 1069 3436 1245 2071 275 1981 4754 2550 3273 558 836 1886 1839 2087 2461 586 2055 1998 3793 1438 1100 1716 4106 4850 1044 4388 1134 1610 3424 1169 3790 1260 1057 4575 2560 3649 261 2837 3743 2752 634 2113 2986 1104 2714 4164 4826 2620 3352 4920 1552 3932 3754 551 2366 3777 4856 1574 4740 4705 68 346 3775 4748 4522 4713 277 3141 3538 1565 1420 3285 3451 3668 3347 2733 4050 1147 1825 3235 395 517 4853 3472 1400 2638 573 1967 3568 3141 1979 3359 3521 2025 3629 4892 3065 2243 4020 3707 1049 757 2663 2701 1403 985 601 1695 4278 2093 3944 767 296 109 4624 1625 1451 3922 1943 3499 987 2466 2522 2821 2303 1607 1952 782 2077 4180 4325 859 3379 3550 722 4984 4124 1751 2107 2492 2007 4905 665 2824 1726 2481 1545 3385 2661 2498 4220 2798 3642 1548 4721 155 109 3963 2944 776 207 5 944 764 4056 1927 1140 1158 571 583 4977 285 2717 263 3139 2982 3310 2272 2208 3326 2262 3702 3517 1952 2388 4851 3911 313 4739 582 3637 1971 2127 3893 3381 4175 2943 648 2828 2678 2962 4024 2770 354 4830 2160 2045 4088 2999 4891 2957 1145 1590 1718 3776 2857 802 999 1878 1135 4171 3495 1664 4663 1300 3644 2052 4732 80 2785 4764 3042 1769 546 496 2562 877 667 1433 2727 2538 2137 1584 1658 1620 3507 4542 73 4020 3313 2389 45 1018 2510 4719 3955 1679 58 1590 1195 2916 4187 1523 4480 1256 550 146 4663 3560 864 3182 1451 3623 4970 97 2032 736 4340 2655 735 3201 2040 124 122 307 3515 3331 385 1543 2560 1377 1907 3539 4550 2745 2960 4477 4130 160 1759 2456 1680 4236 4316 2277 4046 1953 93 2499 2801 1092 2412 4994 1589 2284 1263 4468 711 1397 1527 1893 269 4478 870 3677 1344 4206 3188 4979 8 3195 1843 84 4817 2139 1882 2780 4718 2054 3247 1805 3019 2355 1945 2606 3831 1979 4020 2805 3305 643 3158 288 4233 406 1666 2464 355 1210 1371 96 2672 585 259 3069 294 35 676 3206 4265 2826 4023 2029 2433 86 4907 270 2030 4306 1297 4263 4889 2327 3094 4777 2965 655 3536 3679 3446 4661 4684 3188 233 1187 2471 442 1436 1132 4267 4880 1359 2067 2837 2884 1795 219 2845 2705 4146 3577 1118 730 3528 1607 2350 4968 3600 939 304 1272 2490 1599 2344 3105 2731 2537 1830 3644 4885 4238 1656 2389 2137 635 2457 1489 2402 2596 1574 3424 2836 553 4394 965 1133 3709 1216 3724 738 3416 3681 3299 437 824 3999 788 2134 852 1274 1522 56 993 1112 4106 1894 1992 3002 141 1093 1048 3413 2499 2147 208 3854 3369 2928 1688 3178 2165 3140 1696 2733 2994 3553 2272 1337 4904 2875 2098 1625 4797 4677 869 182 2202 3390 2660 2482 239 788 3311 3883 4953 1359 1735 650 3749 1913 1431 3765 2973 2029 1370 1642 994 1414 944 4781 2370 3545 1461 1060 1597 2557 1171 4073 2099 777 2302 1791 3896 4952 4952 2776 138 3377 3543 2750 2554 2169 4428 2241 2081 21 4478 2481 4814 834 2438 2957 2205 1896 35 643 2457 3047 2569 2800 245 2904 334 874 4560 2591 1069 3620 3431 3568 4951 1546 408 3320 2230 1070 1452 2196 2275 4138 4371 784 2955 4295 3375 3145 1438 3789 2554 1043 265 116 1345 547 4852 3635 3622 3796 661 911 696 4228 566 246 2038 3300 1989 4755 1300 2259 235 3925 1996 2469 3224 4892 1408 3899 733 4036 4471 4754 593 741 3895 4176 936 257 1578 3660 2792 3952 1588 1814 910 2713 2884 2823 1457 3770 633 4146 277 3737 4655 2642 2876 3420 89 998 1127 2270 433 2931 2723 899 182 2889 114 4350 4102 3897 1590 1944 1788 245 718 2396 1955 3680 682 2321 2252 3719 1305 3360 1724 4672 3838 3982 1994 4906 1654 36 71 406 3117 3317 215 3781 898 1872 2039 3458 1784 4431 3000 2065 3942 2863 3971 975 849 3889 4445 1004 904 4928 4452 1935 872 4956 722 3552 3953 3462 2122 2020 4377 3925 1097 2572 2712 1136 4808 2911 1468 3621 2951 2818 3750 570 523 357 2668 2922 4291 2952 872 698 2031 3885 4873 684 362 4024 2184 1563 3802 4228 3630 689 4885 215 2500 574 3071 601 3267 994 3161 1673 4758 2092 2844 785 756 371 4894 1211 2762 3046 4988 4988 3496 4783 4329 872 1955 709 3662 3494 4676 314 2963 4053 3118 4274 1060 415 941 2301 2871 2493 3566 1359 109 3996 2025 3158 2616 1590 4038 4467 4564 1323 535 1837 992 2274 3090 4618 2663 2363 307 2704 2202 3359 2175 3139 952 2322 2779 959 2881 3839 3222 2417 2831 2804 3452 2473 2606 1824 1381 4811 1102 1747 381 2934 3583 3869 4093 435 4172 3922 4451 2570 3266 4867 3189 962 1795 1296 3739 192 2782 3429 454 1667 4466 3339 3147 3245 1043 4881 1221 4044 123 2891 1954 2897 285 3761 452 4352 3966 1803 3868 58 77 3939 4434 4424 44 1699 543 4728 2655 914 2278 1165 341 4505 4288 4138 2357 4586 3136 2791 1787 771 3078 287 3074 3143 4640 3217 4692 718 918 4582 344 1167 3396 4786 4564 2363 509 1959 1371 3403 3318 2550 960 4052 114 4783 2540 2226 1988 3492 3125 3107 3337 2936 4951 1742 2807 3997 4175 3156 257 4295 2806 1301 3477 3435 1626 3734 4327 1853 3827 1634 1255 2584 3701 4742 2232 4622 2537 3785 3555 4242 1881 2004 3988 1876 2708 2375 2554 1014 4878 2684 4909 3853 4452 1023 595 4642 1607 1270 2748 2577 3175 1574 4679 3145 577 2646 84 757 3156 4408 2827 4955 4368 2341 277 179 1392 856 1234 1474 1138 3414 2962 3880 3301 1709 1584 30 58 4274 2124 1036 1358 583 1400 1770 1551 453 3173 2333 2924 1166 4014 3965 2610 4543 1893 1349 2076 1627 2645 2170 235 2771 873 1503 3927 1125 3027 1344 4940 2059 1103 638 3577 3486 3440 2275 3453 2554 1603 4998 3626 3059 539 1961 2734 2344 2517 3697 1856 1592 4269 2782 3552 954 2392 356 1394 1737 188 3216 3853 2784 1700 1295 3823 1498 2041 204 4987 2725 1078 2600 1373 3784 2274 4215 1581 562 1963 968 3851 1813 3337 341 418 2012 4640 2805 4023 1240 2547 2094 1804 153 1533 1654 2112 531 2121 4612 34 2519 4598 491 2766 1196 1075 1655 4559 4021 113 3946 2205 820 760 305 4264 535 2570 2187 4462 419 3800 618 2906 646 3281 1333 1718 1194 1122 748 2650 1798 1299 3294 429 2847 2478 2560 18 1391 2981 4383 2087 3532 4436 4408 3699 1125 3429 3877 846 289 1859 2707 1982 1824 2457 4218 4703 485 2395 677 994 3205 381 3834 777 449 1235 2563 953 1354 577 535 4689 4566 940 2811 1970 2229 3368 1453 2336 1562 3852 3474 729 4558 171 702 3619 3407 3508 4884 634 3225 1108 1975 714 3179 4618 2953 2762 1974 2347 1625 2226 2697 4368 3026 4734 3376 3069 2225 4116 1342 3355 4847 2714 2497 368 1609 347 3045 4013 676 1780 3892 3880 3013 1473 4372 2507 2368 486 3017 4501 1150 2939 1933 1960 2685 4519 1333 1579 3315 160 4198 392 4410 1682 3136 4824 4803 2255 2721 2142 476 2581 2245 857 4283 2006 3757 2838 1264 1572 3637 3846 3574 1091 4906 708 4539 4028 1716 3009 4505 2703 3321 4739 2099 1016 3979 3148 4651 502 450 879 2363 2528 4708 872 4091 2824 4060 4460 4059 2573 2993 3326 4865 1971 1177 4052 3693 606 686 2073 4851 2767 4903 2085 2433 4278 3610 546 3722 3968 2873 1888 1234 1819 4946 2283 11 2696 1635 2667 4215 3520 1326 413 2495 4123 3305 4648 907 1236 4946 4082 2683 4097 3761 2868 1126 1361 1158 2291 3856 1399 4628 3281 3177 1990 2702 4052 588 1413 298 2508 317 246 1651 3179 2671 4882 1939 1600 989 3079 2280 1644 1759 2532 2174 3977 168 643 1830 4256 2861 3720 1214 690 2792 4754 3264 737 334 4792 791 480 1605 4830 3600 296 4803 2608 726 460 2171 2398 3248 735 649 4400 4350 3749 2744 1836 3610 2912 585 3958 3356 124 1407 1999 3001 1964 3569 4673 252 1758 4671 2921 4045 3268 873 1686 2474 2069 2805 3402 4491 2932 1119 4287 890 4735 3853 2059 27 2562 4539 2855 944 2877 1629 2484 1056 2956 44 1687 3281 4037 691 3111 3171 1431 2762 4756 1383 4873 826 4402 3192 4337 3674 3335 2625 3702 1385 3609 3455 4808 4813 2715 793 3926 1723 1842 2653 1768 1016 1955 1404 3159 1496 3631 9 1953 4926 3249 4118 4842 2059 3887 2080 2191 857 944 1779 1636 3346 2886 4049 4617 1925 4099 1647 4696 74 880 793 2058 4781 388 3079 4715 4040 2354 1367 2603 20 1854 4806 2158 4680 2753 2193 3798 2878 2539 2505 968 2865 2001 270 2557 668 3591 4715 3593 183 583 1786 2545 2299 691 921 3512 4857 1791 4598 1211 2205 1819 663 2919 1985 1535 2885 2063 287 1799 1418 1114 422 4402 1128 4259 2338 3798 2596 475 141 1038 795 2549 4338 2055 2243 3040 1140 4631 365 3443 3842 4878 2442 3507 2925 4874 4052 2720 3533 1198 2062 169 1617 3714 1710 3189 3662 585 2050 4413 3691 2510 2449 1454 4866 2465 4052 3770 1896 3201 412 1097 4865 3794 3982 684 502 889 3900 4339 104 4765 3837 447 4938 969 2113 1178 3410 4143 69 926 1928 2135 766 4780 2122 928 4748 390 4559 1702 4260 3271 642 2349 4355 3729 1138 4037 1719 3318 3992 3718 4680 4011 3456 4070 4048 4587 4620 2385 2806 4752 3301 2224 2537 4807 3083 2477 195 1268 3375 1742 4740 1891 279 888 1793 3370 842 1181 1269 3891 2389 4137 3382 3838 901 1098 4235 3678 2997 2014 3341 378 4146 1962 2331 2719 693 3168 418 4524 1017 2872 228 1719 4918 242 1993 4014 1278 4362 1228 3657 4526 4822 1200 2118 2815 2004 474 3244 4053 2991 3912 2668 919 708 794 4764 4400 80 136 1959 1973 3748 181 3650 1137 2268 4274 2591 926 2492 1982 2511 2836 3364 781 3028 3166 1917 1110 4213 1499 1836 4225 4542 2427 4695 4656 2255 2924 782 462 4587 3619 2668 4598 340 318 4845 2653 4172 3921 4658 530 951 1255 611 2773 4960 2740 605 2262 2042 553 3324 3143 2064 2600 330 653 4870 952 4937 2738 409 3451 3101 4023 1116 4278 1878 1377 3581 3984 4053 1649 72 2991 1204 912 2219 744 4707 4434 1089 3469 3256 1459 3282 484 894 4367 2354 4141 2162 4702 3337 4084 3162 2313 4300 1872 3830 550 2627 3259 4044 658 1034 4452 807 1935 1627 3201 3807 2603 243 2400 2919 4659 460 3016 4843 1392 2820 187 1400 3786 4804 2719 601 1634 2115 158 1654 508 1260 2938 408 4493 597 1317 1107 1737 3689 1256 3827 592 1761 2074 698 2393 4542 3948 2504 3400 3907 4986 4316 1581 4920 4980 578 4701 1849 3230 4558 2980 4185 4869 3560 2330 3353 235 2049 3142 3126 1709 4164 1 2962 3187 4928 1237 3993 455 1613 3141 4390 3706 1901 4910 247 3307 662 2535 1391 617 541 2157 322 1020 789 4426 4494 2126 4600 3135 3873 3712 557 1279 1765 2402 3274 2462 3187 1531 4548 609 879 2482 637 1456 4328 796 2525 1706 3691 1655 4164 1391 40 710 94 1580 1509 1110 1289 1163 1313 1000 1147 2649 1692 2889 3888 2036 717 4590 1626 180 2277 4492 793 4128 2154 126 3550 4861 4534 1112 2706 963 153 3380 3044 2905 3020 3347 131 834 48 199 1489 2879 1326 2086 945 4551 4392 1097 1738 700 1391 574 899 2768 703 4098 2994 4656 3072 795 2240 3080 2231 1358 1919 823 4948 964 2399 453 3893 3587 2309 4586 2083 3402 181 4832 4002 3708 4482 403 2791 1377 3275 3608 630 3937 4216 364 754 714 2629 4646 1520 3788 870 3304 3814 4827 1461 1946 879 261 671 2624 3746 1213 2821 1862 3021 4127 4098 4760 2180 1284 728 342 1807 2144 4057 3496 162 2915 4226 4494 4718 3237 1458 4396 1736 2302 636 3012 488 4437 4352 700 629 1934 1113 3785 1904 2481 1699 3092 326 113 3163 828 1090 4650 2924 4412 1178 2647 1175 3383 504 4627 2114 385 900 4494 2036 4968 1415 986 232 827 3715 2951 1096 3977 4786 3483 3573 262 3597 254 2668 385 1783 4734 4706 1785 1512 1662 1411 3465 4541 2460 1447 4713 1856 1243 1567 1064 1572 1598 2598 567 2981 382 4835 4439 557 1966 647 3910 1193 4532 658 2958 3638 795 1858 4294 2246 983 3359 2125 639 4665 2704 2524 4686 3892 1154 4243 319 1383 3147 255 4537 4084 2650 2403 1784 2879 721 524 146 43 4573 2721 3391 1354 4782 2235 4522 2068 2002 906 2893 3218 1542 576 3372 291 448 4378 4001 2270 983 4075 96 953 3116 955 251 4065 4541 4369 321 2619 2342 244 1458 2283 1283 377 86 1179 45 3280 775 3519 3257 1175 1428 4254 3959 3267 408 4075 2080 1756 1434 3316 1905 1185 1958 2947 4720 3020 2180 4060 1221 1327 3007 2342 3275 639 2977 2781 2287 4647 1813 3319 2784 2168 733 1102 1151 4640 1013 1733 4825 3033 392 2775 1874 1108 2121 4836 3517 2365 4020 3205 1185 4857 1887 2642 614 2181 1139 2842 1249 2336 1679 491 236 2484 3767 2123 281 4153 4808 302 8 2027 4944 2432 2975 4597 172 165 3362 1069 4202 2095 1003 3258 1160 4417 27 243 2044 1066 3242 3370 261 2491 4828 3826 306 644 629 4388 1698 2188 830 96 363 1360 2879 1290 4223 3950 3559 4411 4452 4274 3235 2042 3107 4017 3419 2842 3764 892 2835 3934 4431 2431 953 3442 726 963 266 2627 1337 2688 2362 2450 2522 2367 3587 4453 4371 1863 821 2679 1564 4790 928 4722 1457 147 4030 4073 2739 2802 1156 2138 278 775 410 2428 1003 3342 521 2542 2063 2218 4316 2073 124 2533 2051 2046 4926 1928 2036 3938 2928 2427 1458 1201 4688 4935 3579 1345 4937 2706 2052 3561 2568 875 1872 2425 4945 2318 3745 803 3658 3010 4350 259 2472 2673 2317 1724 1649 56 427 635 1002 4704 2154 3311 2194 1788 122 2609 230 789 3700 2400 3054 2898 1954 80 4192 1163 547 2172 3428 719 4222 3850 3936 1805 1156 303 4517 3976 4931 2458 1609 647

45954

**Case = 2:**

41684

**Case = 3:**

500000000

**Case = 4:**

48932400

**Case = 5:**

558491184516

/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include <iostream>

#include <stack>

#include <vector>

using namespace std;

// Hàm tính diện tích hình chữ nhật lớn nhất từ các cột

long long largestRectangleArea(vector<long long>& heights) {

stack<long long> st; // Stack lưu trữ chỉ số các cột tăng dần

long long maxArea = 0; // Biến lưu trữ diện tích lớn nhất tìm được

long long i = 0; // Chỉ số cột hiện tại

while (i < heights.size()) {

if (st.empty() || heights[i] >= heights[st.top()]) {

// Nếu stack rỗng hoặc chiều cao của cột hiện tại lớn hơn hoặc bằng cột trên đỉnh stack

// Đẩy chỉ số cột hiện tại vào stack và tăng chỉ số cột

st.push(i);

i++;

} else {

// Nếu chiều cao của cột hiện tại nhỏ hơn cột trên đỉnh stack

// Lấy chiều cao của cột trên đỉnh stack và tính diện tích hình chữ nhật

long long tp = st.top();

st.pop();

long long area = heights[tp] \* (st.empty() ? i : i - st.top() - 1);

maxArea = max(maxArea, area);

}

}

// Kiểm tra các cột còn lại trong stack

while (!st.empty()) {

long long tp = st.top();

st.pop();

long long area = heights[tp] \* (st.empty() ? i : i - st.top() - 1);

maxArea = max(maxArea, area);

}

return maxArea;

}

int main() {

long long n;

cin >> n;

vector<long long> heights(n);

for (long long i = 0; i < n; i++) {

cin >> heights[i];

}

long long result = largestRectangleArea(heights);

cout << result << endl;

return 0;

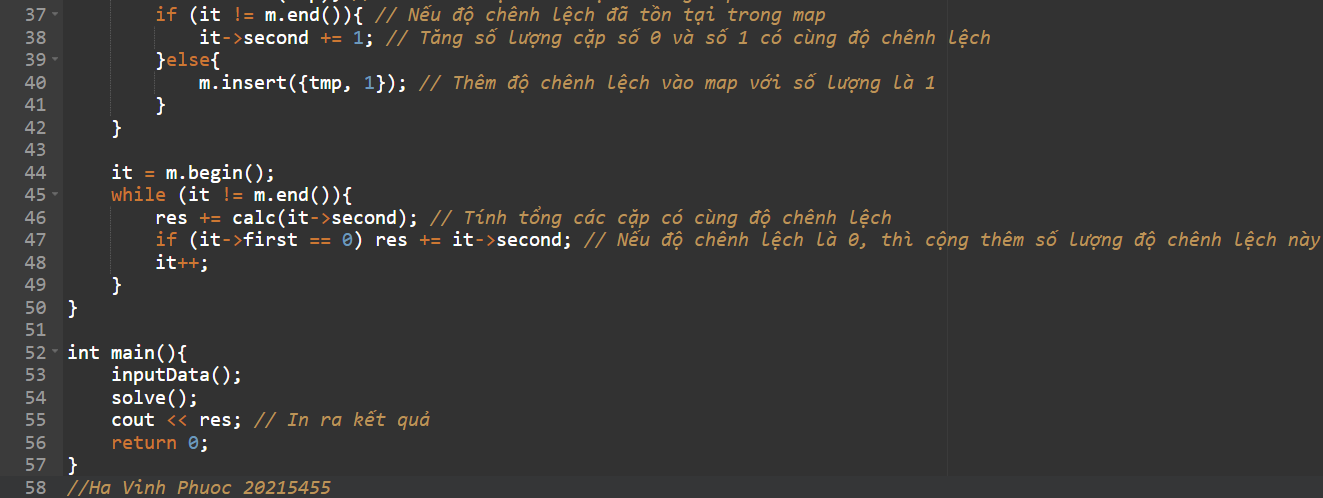
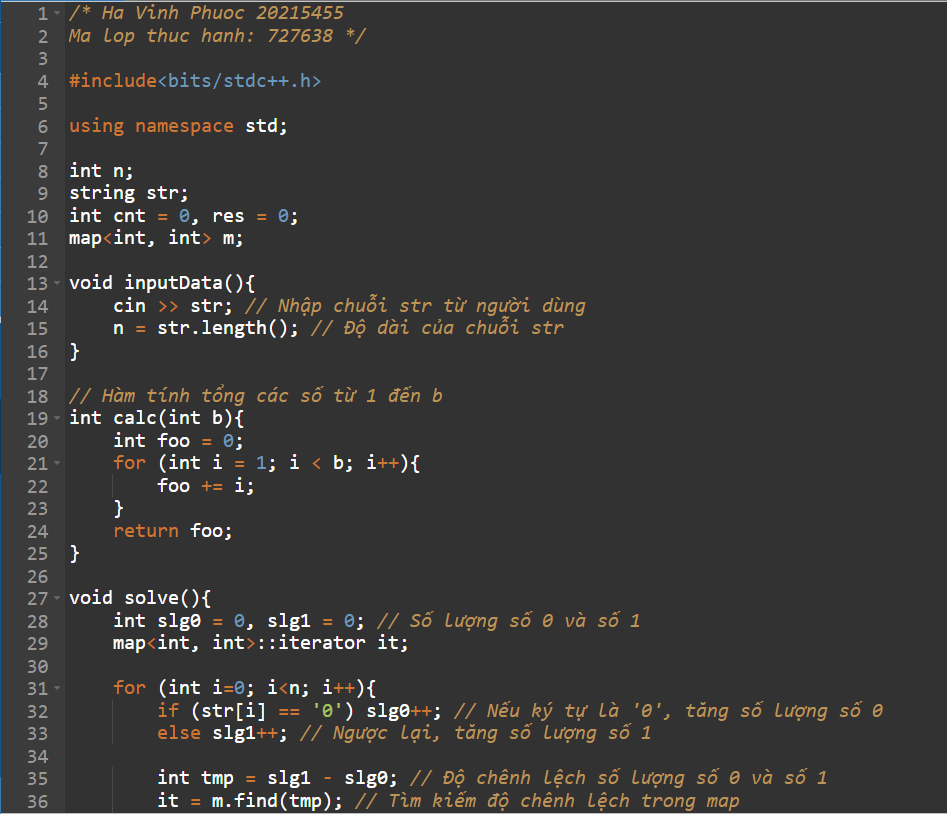
}

//Ha Vinh Phuoc 20215455

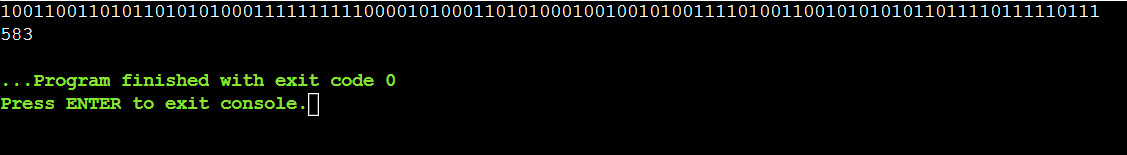
## **Bài tập 4.13.**

## Cho một xâu nhị phân độ dài n.

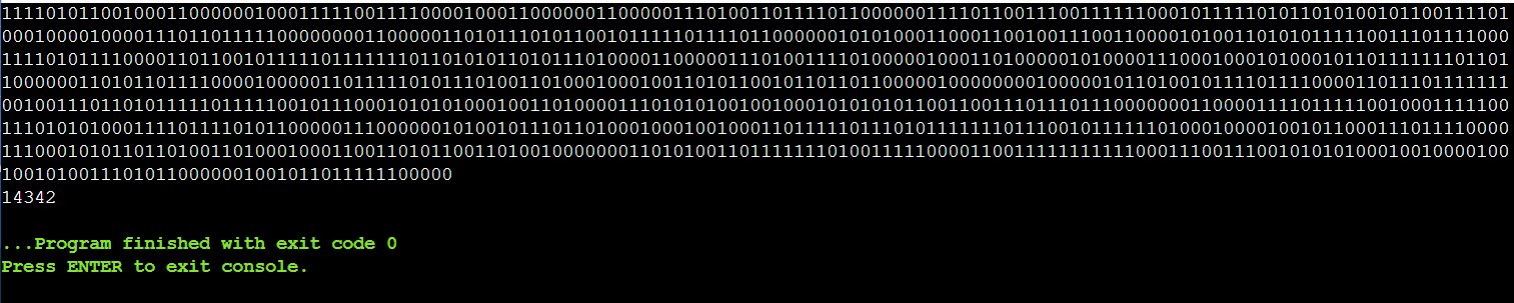
## Hãy viết chương trình đếm số lượng xâu con chứa số ký tự 0 và số ký tự 1 bằng nhau.



**Case = 1:**



**Case = 2:**



**Case = 3:**

10354354

**Case = 4:**

281665459

**Case = 5:**

1105947673

/\* Ha Vinh Phuoc 20215455

Ma lop thuc hanh: 727638 \*/

#include<bits/stdc++.h>

using namespace std;

int n;

string str;

int cnt = 0, res = 0;

map<int, int> m;

void inputData(){

cin >> str; // Nhập chuỗi str từ người dùng

n = str.length(); // Độ dài của chuỗi str

}

// Hàm tính tổng các số từ 1 đến b

int calc(int b){

int foo = 0;

for (int i = 1; i < b; i++){

foo += i;

}

return foo;

}

void solve(){

int slg0 = 0, slg1 = 0; // Số lượng số 0 và số 1

map<int, int>::iterator it;

for (int i=0; i<n; i++){

if (str[i] == '0') slg0++; // Nếu ký tự là '0', tăng số lượng số 0

else slg1++; // Ngược lại, tăng số lượng số 1

int tmp = slg1 - slg0; // Độ chênh lệch số lượng số 0 và số 1

it = m.find(tmp); // Tìm kiếm độ chênh lệch trong map

if (it != m.end()){ // Nếu độ chênh lệch đã tồn tại trong map

it->second += 1; // Tăng số lượng cặp số 0 và số 1 có cùng độ chênh lệch

}else{

m.insert({tmp, 1}); // Thêm độ chênh lệch vào map với số lượng là 1

}

}

it = m.begin();

while (it != m.end()){

res += calc(it->second); // Tính tổng các cặp có cùng độ chênh lệch

if (it->first == 0) res += it->second; // Nếu độ chênh lệch là 0, thì cộng thêm số lượng độ chênh lệch này

it++;

}

}

int main(){

inputData();

solve();

cout << res; // In ra kết quả

return 0;

}

//Ha Vinh Phuoc 20215455