ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

**BÁO CÁO BÀI THỰC HÀNH SỐ 4**

**MÔN KỸ THUẬT LẬP TRÌNH**

Sinh viên: Lê Đình Trí Tuệ

MSSV: 20210909

Mã lớp: 732833

Contents

[Bài thực hành số 4 – Tuần 16 6](#_Toc154693781)

[**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). 6](#_Toc154693782)

[**Bài tập 4.2.** Một điểm trong không gian 2 chiều được biểu diễn bằng pair. Hãy viết hàm tính diện tích tam giác theo tọa độ 3 đỉnh. 9](#_Toc154693783)

[**Bài tập 4.3.** Một vector trong không gian 3 chiều được biểu diễn bằng tuple<double, double, double>. Hãy viết hàm tính tích có hướng của 2 vector. 10](#_Toc154693784)

[**Bài tập 4.4.** Cho hai std::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. 12](#_Toc154693785)

[**Bài tập 4.5.** Viết hàm void dfs(vector< list<int> > adj) thực hiện thuật toán DFS không sử dụng đệ quy. 15](#_Toc154693786)

[**Bài tập 4.6.** Viết hàm void dfs(vector< list<int> > adj) thực hiện thuật toán BFS không sử dụng đệ quy.. 17](#_Toc154693787)

[**Bài tập 4.7.** Viết các hàm thực hiện các phép giao và hợp của hai tập hợp được biểu diễn bằng set. 18](#_Toc154693788)

[**Bài tập 4.8.** Viết các hàm thực hiện các phép giao và hợp của hai tập hợp mờ được biểu diễn bằng map. 19](#_Toc154693789)

[**Bài tập 4.9.** Cài đặt thuật toán Dijkstra trên đồ thị vô hướng được biểu diễn bằng danh sách kề sử dụng std::priority\_queue. 20](#_Toc154693790)

[**Bài tập 4.10.** Xây dựng một máy tìm kiếm (search engine) đơn giản. 21](#_Toc154693791)

[**Bài tập 4.11.** Hãy viết chương trình phân bố lính đứng ở các đoạn tường sao cho tổng số lính là s và tổng số lượng tên giặc lọt vào lâu đài là nhỏ nhất. 27](#_Toc154693792)

[**A screen shot of a computer

Description automatically generated** 31](#_Toc154693793)

[A blue screen with white text

Description automatically generated 32](#_Toc154693794)

[**Bài tập 4.12.** 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. 33](#_Toc154693795)

[**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. 36](#_Toc154693796)

Figure 1 4

Figure 2 4

Figure 3 5

Figure 4 5

Figure 5 6

Figure 6 7

Figure 7 7

Figure 8 8

Figure 9 9

Figure 10 10

Figure 11 10

Figure 12 11

Figure 13 11

Figure 14 12

Figure 15 13

Figure 16 14

Figure 17 15

Figure 18 16

Figure 19 16

Figure 20 17

Figure 21 17

Figure 22 18

Figure 23 18

Figure 24 19

Figure 25 23

Figure 26 24

Figure 27 25

Figure 28 29

Figure 29 30

Figure 30 31

Figure 31 34

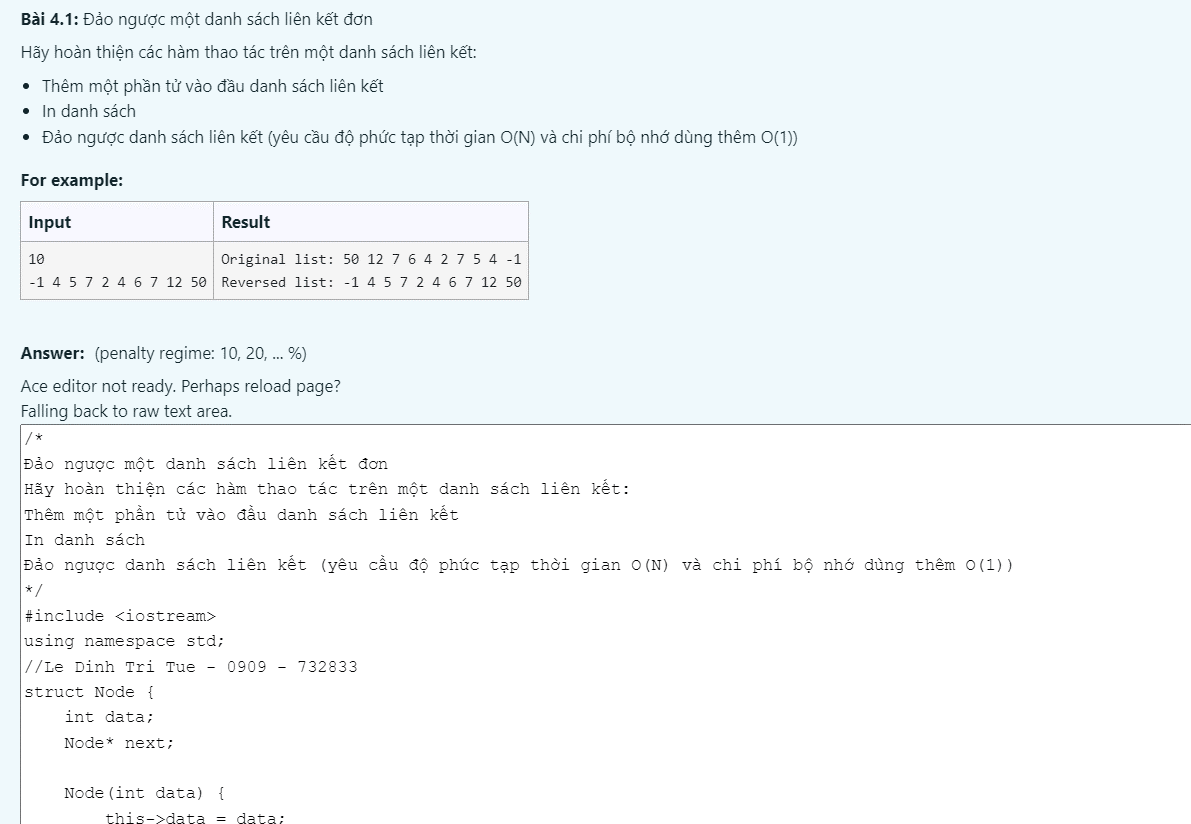
Figure 32 36

Figure 33 36

# Bài thực hành số 4 – Tuần 16

**Copy code (kèm ảnh code và kết quả vào )**

## **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).



Figure

A screenshot of a computer

Description automatically generated

Figure

A white background with black text

Description automatically generated

Figure

A white background with black text

Description automatically generated

Figure

A screenshot of a computer

Description automatically generated

Figure

## **Bài tập 4.2.** Một điểm trong không gian 2 chiều được biểu diễn bằng pair. Hãy viết hàm tính diện tích tam giác theo tọa độ 3 đỉnh.

A screenshot of a computer

Description automatically generated

Figure

A screenshot of a computer

Description automatically generated

Figure

## **Bài tập 4.3.** Một vector trong không gian 3 chiều được biểu diễn bằng tuple<double, double, double>. Hãy viết hàm tính tích có hướng của 2 vector.

A screenshot of a computer

Description automatically generated

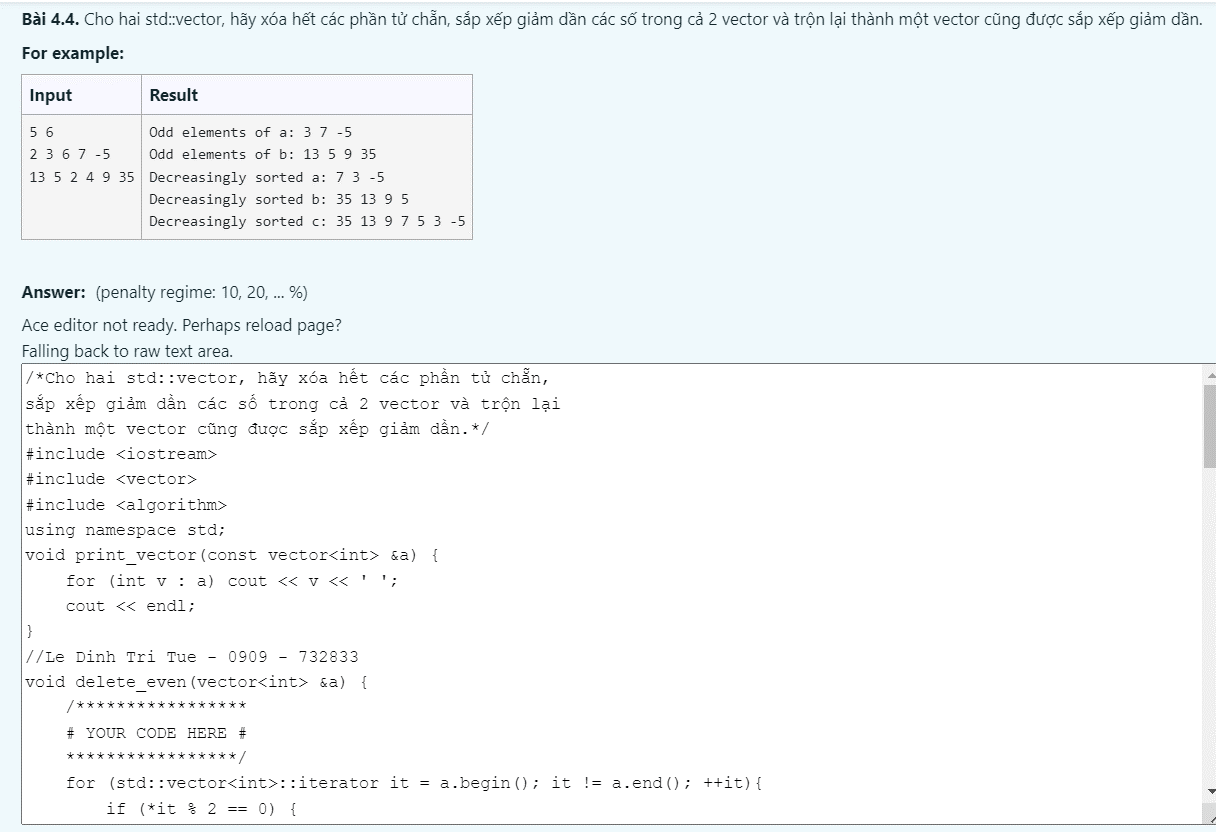
Figure

A screenshot of a computer program

Description automatically generated

Figure

## **Bài tập 4.4.** Cho hai std::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.



Figure

A white background with black text

Description automatically generated

Figure

A screenshot of a computer

Description automatically generated

Figure

A screenshot of a computer

Description automatically generated

Figure

A screenshot of a computer

Description automatically generated

Figure

## **Bài tập 4.5.** Viết hàm void dfs(vector< list<int> > adj) thực hiện thuật toán DFS không sử dụng đệ quy.

A screenshot of a computer

Description automatically generated

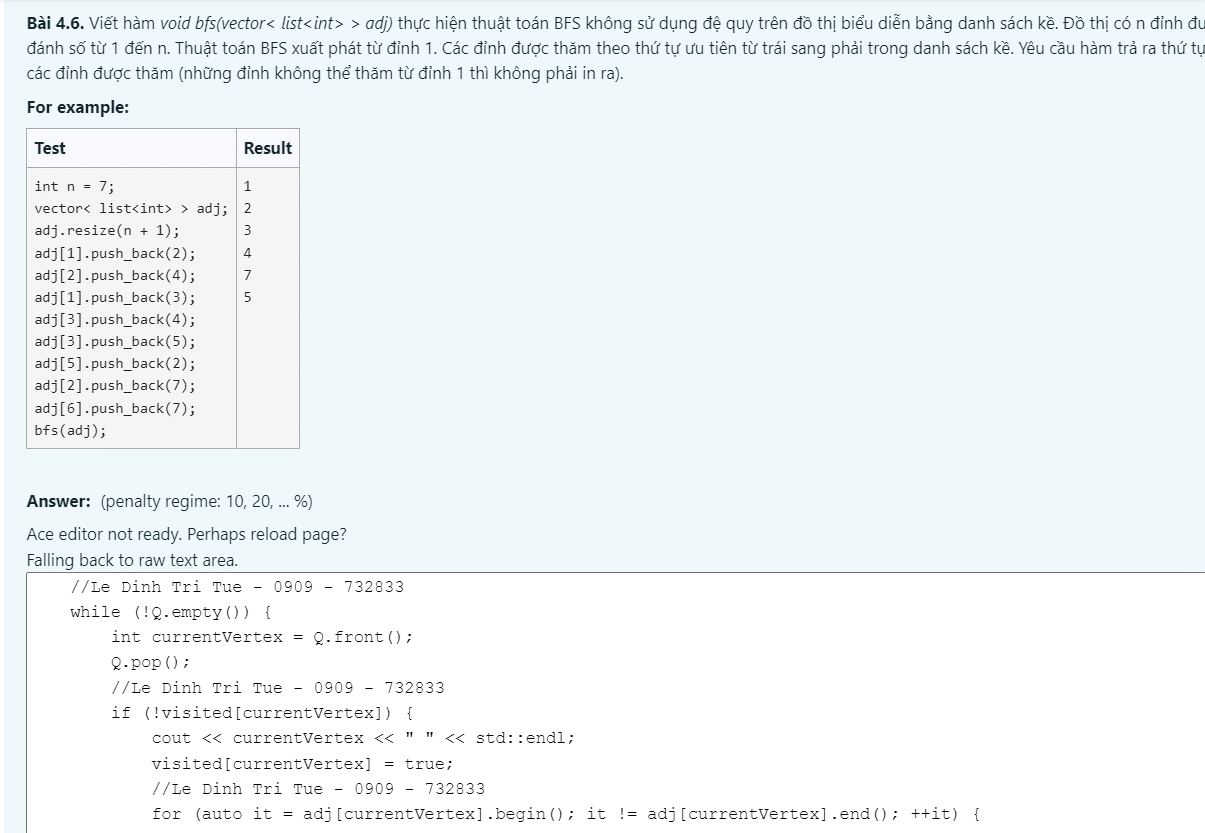
Figure

A screenshot of a computer

Description automatically generated

Figure

## **Bài tập 4.6.** Viết hàm void dfs(vector< list<int> > adj) thực hiện thuật toán BFS không sử dụng đệ quy..



Figure

A screenshot of a computer

Description automatically generated

Figure

## **Bài tập 4.7.** Viết các hàm thực hiện các phép giao và hợp của hai tập hợp được biểu diễn bằng set.

A screenshot of a computer

Description automatically generated

Figure

A screenshot of a computer

Description automatically generated

Figure

## **Bài tập 4.8.** Viết các hàm thực hiện các phép giao và hợp của hai tập hợp mờ được biểu diễn bằng map.

A close-up of a text

Description automatically generated

Figure

A screenshot of a computer program

Description automatically generated

Figure

## **Bài tập 4.9.** Cài đặt thuật toán Dijkstra trên đồ thị vô hướng được biểu diễn bằng danh sách kề sử dụng std::priority\_queue.

A white paper with black text

Description automatically generated

Figure

A screenshot of a computer

Description automatically generated

Figure

## **Bài tập 4.10.** Xây dựng một máy tìm kiếm (search engine) đơn giản.

**CODE:**

/\*Xây dựng một máy tìm kiếm (search engine) đơn giản.

Cho N văn bản và Q truy vấn. Với mỗi truy vấn, cần trả về

văn bản khớp với truy vấn đó nhất. Sử dụng phương pháp tính điểm TF-IDF:\*/

#include<bits/stdc++.h>

using namespace std;

#define N 10000

vector<string> doc[N];

map<string, int> f[N];

map<string, int> df;

int maxf[N], n;

//Le Dinh Tri Tue - 0909 - 732833

double TF(string t, int d){

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

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

}//Le Dinh Tri Tue - 0909 - 732833

double IDF(string t){

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

}

double score(string t, int d){

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

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

}//Le Dinh Tri Tue - 0909 - 732833

int main(){

ios\_base::sync\_with\_stdio(0);

cin.tie(0);

cout.tie(0);

//Le Dinh Tri Tue - 0909 - 732833

cin >> n;

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

string t;

cin >> t;

string w;

map<string, int> d\_f;

//Le Dinh Tri Tue - 0909 - 732833

for(char c : t){

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

else {

d\_f[w]++;

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

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

f[i][w]++;

w = "";

continue;

}

}

d\_f[w]++;

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

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

f[i][w]++;

for(auto it : f[i]){

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

}

}

//Le Dinh Tri Tue - 0909 - 732833

int q;

cin >> q;

while(q--){

vector<string> query;

string t;

cin >> t;

string w;

for(char c : t){

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

else{

query.push\_back(w);

w = "";

continue;

}

}

//Le Dinh Tri Tue - 0909 - 732833

query.push\_back(w);

double tempscore = -1e9;

int document = 0;

int d = 0;

while (d < n){

double totalscore = 0;

for(string t : query){

totalscore += score(t, d);

}

d++;

if(totalscore > tempscore){

tempscore = totalscore;

document = d;

}

}

cout << document << "\n";

}

}

**OUTPUT:**

A screen shot of a computer

Description automatically generated

Figure

A screenshot of a computer

Description automatically generated

Figure

A screenshot of a computer

Description automatically generated

Figure

## **Bài tập 4.11.** Hãy viết chương trình phân bố lính đứng ở các đoạn tường sao cho tổng số lính là s và tổng số lượng tên giặc lọt vào lâu đài là nhỏ nhất.

**CODE:**

/\*

Bức tường bao quanh một lâu đài nọ được cấu thành từ n đoạn tường được đánh số từ 1 đến n

. Quân giặc lên kế hoạch tấn công lâu đài bằng cách gửi ai tên giặc đánh vào đoạn tường thứ i.

Để bảo vệ lâu đài có tất cả s lính. Do các đoạn tường có chất lượng khác nhau nên khả năng

bảo vệ tại các đoạn tường cũng khác nhau. Cụ thể tại đoạn tường thứ i , mỗi lính có thể đẩy

lùi tấn công của ki tên giặc.

Giả sử đoạn tường thứ i có xi lính. Khi đó nếu số tên giặc không vượt quá xi×ki thì không có

tên giặc nào lọt vào được qua đoạn tường này. Ngược lại sẽ có ai−xi×ki tên giặc lọt vào lâu đài

qua đoạn tường này. Yêu cầu hãy viết chương trình phân bố lính đứng ở các đoạn tường sao cho tổng số

lính là s và tổng số lượng tên giặc lọt vào lâu đài là nhỏ nhất.\*/

#include<bits/stdc++.h>

using namespace std;

//Le Dinh Tri Tue - 0909 - 732833

void algo(){

while(!now.empty() && s > 0){

wall a = now.top();now.pop();

if(a.ai <= a.ki){

dich\_da\_giet += a.ai;

} else {

int now\_enemy = a.ai - a.ki;

now.push(wall(now\_enemy,a.ki));

dich\_da\_giet += a.ki;

}

s -= 1;

}

}

//Le Dinh Tri Tue - 0909 - 732833

struct wall{

int ai;

int ki;

wall(int ai,int ki){

this->ai = ai;

this->ki = ki;

}

};

//Le Dinh Tri Tue - 0909 - 732833

struct compare{

bool operator() (wall a, wall b){

int ra, rb;

if(a.ai <= a.ki) ra = a.ai;

else ra = a.ki;

if(b.ai <= b.ki) rb = b.ai;

else rb = b.ki;

return ra < rb;

}

};

//Le Dinh Tri Tue - 0909 - 732833

int n,s;

priority\_queue<wall,vector<wall>,compare> now;

int tong\_dich=0,dich\_da\_giet=0;

void input(){

cin >> n >> s;

int i = 0;

while(i<n){

int ai,ki;

cin >> ai >> ki;

now.push(wall(ai,ki));

tong\_dich += ai;

i++;

}

}

//Le Dinh Tri Tue - 0909 - 732833

int main(){

input();

algo();

cout << tong\_dich - dich\_da\_giet;

}

**OUTPUT:**

## **A screen shot of a computer Description automatically generated**

Figure

## A blue screen with white text Description automatically generated

Figure

A screenshot of a computer

Description automatically generated

Figure

## **Bài tập 4.12.** 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.

**CODE:**

/\*

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.

\*/

#include<bits/stdc++.h>

using namespace std;

//Le Dinh Tri Tue - 0909 - 732833

int n;

vector<int> graph;

vector<vector<int>> neighbor;

vector<int> value;

void input(){

cin >> n;

graph.push\_back(-1);//dat linh canh

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

int tmp;

cin >> tmp;

graph.push\_back(tmp);

}

graph.push\_back(-1);//linh canh

}

//Le Dinh Tri Tue - 0909 - 732833

int main(){

input();

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

int part = graph.at(i);

vector<int> a;

a.push\_back(part);

//Le Dinh Tri Tue - 0909 - 732833

int now = i;

while(true){

if(graph.at(now - 1) < part){

break;

} else {

a.push\_back(graph.at(now - 1));

now -= 1;

}

}

now = i;

while(true){

if(graph.at(now + 1) < part){

break;

} else {

a.push\_back(graph.at(now + 1));

now += 1;

}

}

value.push\_back(part\*a.size());

}

//Le Dinh Tri Tue - 0909 - 732833

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

cout << value.at(0);

}

**OUTPUT:**

****

Figure

## **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.

**CODE:**

/\*

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.

\*/

#include<bits/stdc++.h>

using namespace std;

const int MAX = 1000000;

string str;

int n, cnt;

int slg0[MAX]; // slg0[i] la dem so luong so 0 trong khoang 0 toi i

int slg1[MAX]; // slg0[i] la dem so luong so 1 trong khoang 0 toi i

//Le Dinh Tri Tue - 0909 - 732833

void input(){

cnt = 0;

cin >> str;

n = str.length();

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

if(i==0){

if(str[i] == '0'){

slg0[i] = 1;

slg1[i] = 0;

} else {

slg0[i] = 0;

slg1[i] = 1;

}

continue;

}

//Le Dinh Tri Tue - 0909 - 732833

if(str[i] == '0'){

slg0[i] = slg0[i-1] + 1;

slg1[i] = slg1[i-1];

} else {

slg0[i] = slg0[i-1];

slg1[i] = slg1[i-1] + 1;

}

}

}

//Le Dinh Tri Tue - 0909 - 732833

void solve(){

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

for(int j=i+1; j<n; j = j + 2){

if(i==0){

if(slg0[j] == slg1[j]) cnt++;

} else {

int c0 = slg0[j] - slg0[i-1];

int c1 = slg1[j] - slg1[i-1];

if(c0 == c1) cnt++;

}

}

}

}

//Le Dinh Tri Tue - 0909 - 732833

int main(){

ios\_base::sync\_with\_stdio(false); cin.tie(NULL);

input();

solve();

cout << cnt;

return 0;

}

**OUTPUT:**

**A screen shot of a computer code

Description automatically generated**

Figure

A blue background with white and blue text

Description automatically generated with medium confidence

Figure