A Project Report
On
DELHI METRO
For
Algorithm and Problem Solving Lab (15B17CI471)

Submitted by:

Apoorv Aggarwal
(9921103094)

Submitted to:

Dr. Neeraj Jain

Dr. Laxmi Chaudhary

Department of CSE/IT

Jaypee Institute of Information Technology University, Noida

May, 2023

Problem Statement

Travel The Delhi Metro is a mass rapid transit (MRT) system serving. Delhi and its satellite cities of Ghaziabad, Faridabad, Gurgaon, Noida, Bahadurgarh and Ballabhgarh, in the National Capital Region of India. The network consists of several color-coded lines serving many stations with a total length of 348.12 km. It is by far the largest and busiest metro rail system in India. Delhi Metro operates over 2700 trips daily, starting at around 05:00 and ending at 23:30. Annual ridership of Delhi Metro can be as high as 1.79 billion.

Delhi metro consist of 7 different colored metro lines including blueline, orange line, yellow line, violet line, red line green line and blue extension line and having a total of 138 metro stations. It is not easy to find which metro station lies on which metro line.

Introduction

Motivation

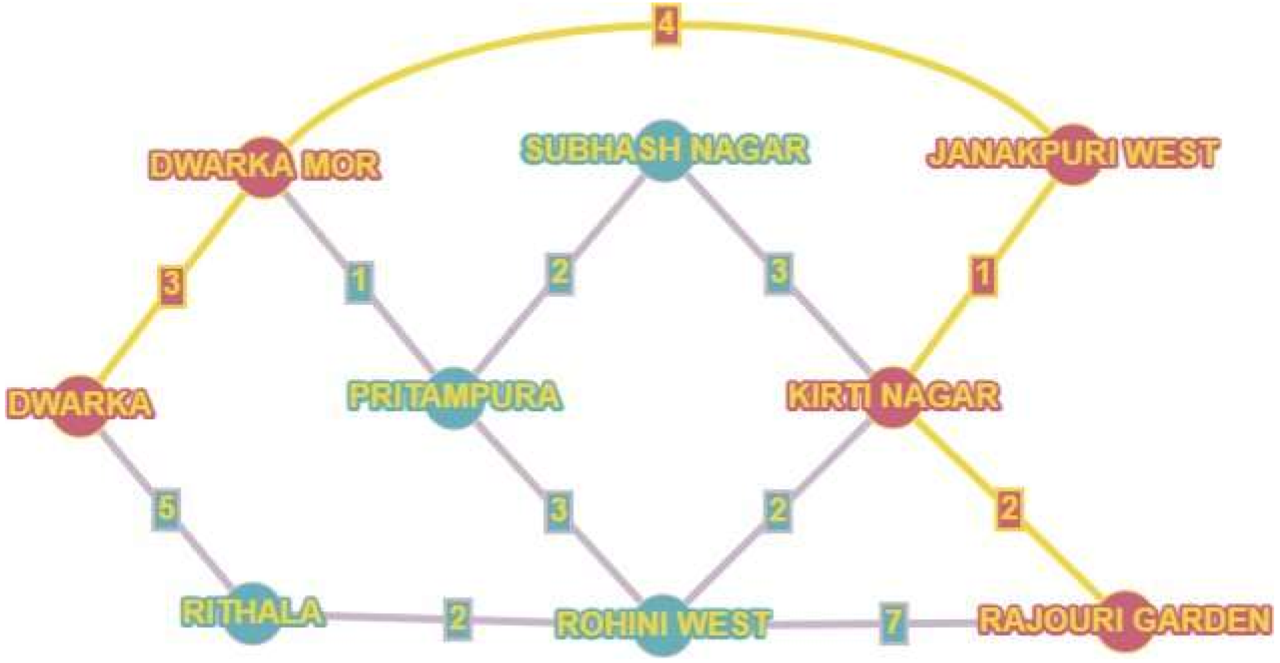
We have developed a metro management system that uses three algorithms to optimize thetravel time and payment process for metro passengers. Firstly, we use backtracking to find all possible paths between the starting and destination stations. Secondly, we use Dijkstra's algorithm to identify the shortest path among the possible routes. By combining these algorithms, our system can provide efficient and reliable solutions for passengers using themetro network.

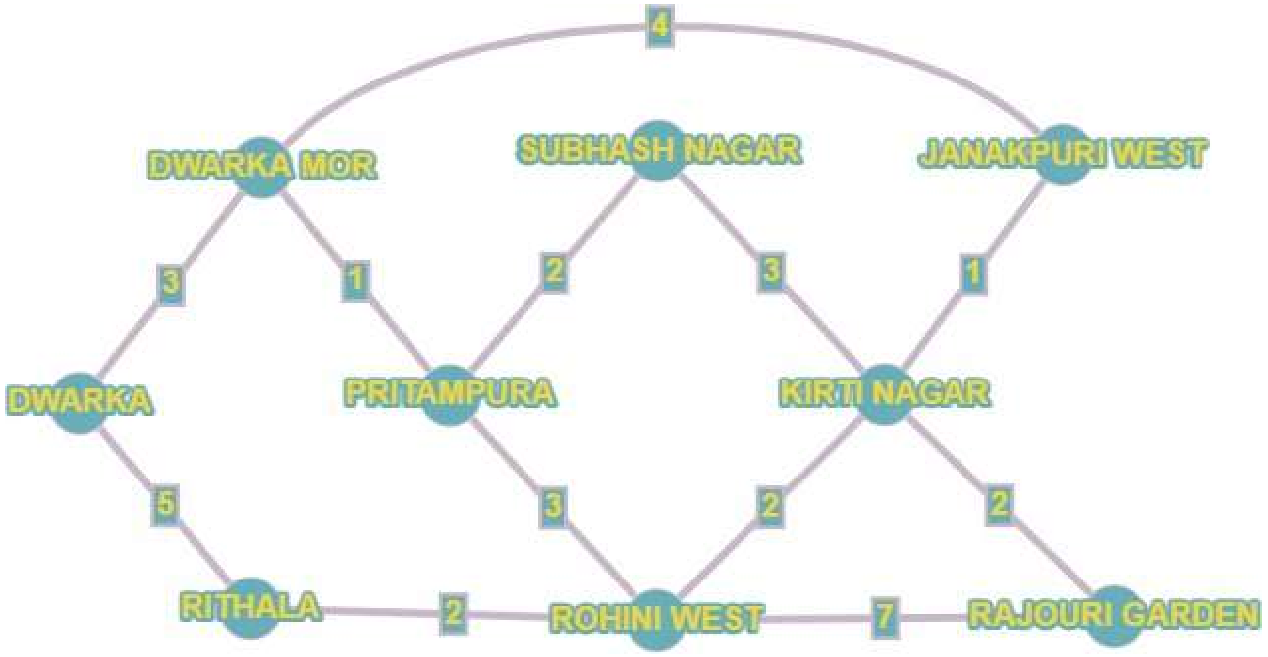
Objective

- To improve the efficiency of the metro system by reducing travel time and increasingcapacity.
- To improve the reliability of the metro system by minimizing delays and improving the accuracy of train schedules.
- To provide passengers with an optimized travel route that minimizes the time anddistance required for their journey, while considering various factors such as congestion, transfers between lines, and train frequency.
- To increase the safety of the metro system by reducing the risk of accidents and improving emergency response times.
- To improve the overall user experience of the metro system by providing passengers with accurate and up-to-date information about train schedules, delays, and other relevant information.
- To reduce the environmental impact of the metro system by optimizing travel routes to minimize energy consumption and reduce carbon emissions.
- To reduce the operating costs of the metro system by optimizing train schedules androutes, reducing maintenance costs, and increasing overall efficiency.

Description of the project

We have developed a metro management system that uses three algorithms to optimize thetravel time and payment process for metro passengers. Firstly, we use backtracking to find all possible paths between the starting and destination stations. Secondly, we use Dijkstra's algorithm to identify the shortest path among the possible routes. By combining these algorithms, our system can provide efficient and reliable solutions for passengers using themetro network.





Implementation
tourplace.txt

India Gate

Central Secratariat

Connaught Place Rajiv

Chowk Lodhi Gardens

Jor Bagh Purana

Quila Pragati

MaidanSansad

Bhavan

Central SecratariatRed

Fort

Chandni Chowk

Salimgarh Fort

Kashmere Gate

Chandni Chowk

Chandni Chowk

Safdarjung's TombJor

Bagh

Qutab Minar

Qutab Minar

Tughlakabad

Tughlakabad

Akshardham Temple

Akshardham

Birla Mandir

R K Ashram Marg

Cathedral Church of RedemptionCentral

Secratariat

Gurdwara Bangla SahibRajiv

Chowk

ISKCON Temple

Kalkaji Mandir Jama

Masjid Chandni

Chowk Lotus Temple

Kalkaji Mandir St.

James' Church

Kashmere Gate

Kalkaji Mandir

Kalkaji Mandir

National Museum

Udyog Bhawan

National Rail Museum

Mandi House

Jantar Mantar

Patel Chowk

Nizamuddin DargahJLN

Stadium

Raj Ghat Chandni

ChowkShanti Vana

Chandni Chowk

National Zoological ParkPragati

Maidan Rashtrapati Bhavan

Central Secratariat

list.txt

```
Jahangirpuri
Adarsh Nagar
Azadpur Model
Town GTB
Nagar
Vishwa Vidyalaya
Vidhan Sabha Civil
Lines Kashmere Gate
Chandni Chowk
Chawri Bazar New
Delhi
Rajiv  Chowk Patel
Chowk Central
SecretariatUdyog
Bhawan Race Course
Jor Bagh
INA
AIIMS
Green Park Hauz
Khas Malviya
NagarSaket
Qutub Minar
Chhatarpur
Sultanpur
Ghitorni
Arjan Garh
Guru Dronacharya
Sikandarpur
MG Road
HUDA City Centre
Noida City Centre
Noida Golf Course
Botanical Garden Noida
Sector 18
Noida Sector 16
Noida Sector 15 New
Ashok Nagar
Mayur Vihar ExtensionMayur
Vihar-I Akshardham
Yamuna Bank
Indraprastha Pragati
Maidan Mandi House
Barakhamba RoadR K
Ashram Marg
Jhandewalan Karol
Bagh Rajendra Place
Patel Nagar Shadipur
Moti Nagar Ramesh
Nagar Rajouri Garden
```

Tagore Garden
Subhash Nagar
Janakpuri East
Janakpuri West Uttam
Nagar East Uttam
Nagar WestNawada
Dwarka Mor
Dwarka
Dwarka Sector 14
Dwarka Sector 13
Dwarka Sector 12
Dwarka Sector 11
Dwarka Sector 10
Dwarka Sector 9
Dwarka Sector 8
Dwarka Sector 21
Khan Market JLN
Stadium Jangpura
Lajpat Nagar
Moolchand Kailash
ColonyNehru Place
Kalkaji Mandir
Govind Puri Okhla
Jasola Apollo
Sarita Vihar

Mohan Estate
Tughlakabad
Badarpur Laxmi
Nagar Nirman
ViharPreet Vihar
Karkarduma
Anand Vihar ISBT
Kaushambi Vaishali
Kirti Nagar
Inderlok
Satguru Ramsingh MargAshok
Park Main Punjabi Bagh East
Shivaji Park
Madipur
Paschim Vihar East
Paschim Vihar WestPeera
Garhi
Udyog Nagar
Surajmal Stadium
Nangloi
Nangloi Railway station
Rajdhani Park
Mundka Dilshad
GardenJhilmil
Mansarovar Park

Delhi Shahdara

Welcome

Seelampur Shastri

Park Tis Hazari

Pul Bangash

Pratap Nagar

Shastri Nagar

Kanhiya Nagar

Keshav Puram

Netaji Subhash Place

Kohat Enclave Pitam

Pura

Rohini East

Rohini West

Rithala

Shivaji Stadium

Dhaula Kuan Delhi

Aerocity Airport

Palam Vihar

Maruti Udyog

IFFCO Chowk

blueline.txt

Noida City Centre

Noida Golf Course

Botanical Garden Noida

Sector 18

Noida Sector 16

Noida Sector 15 New

Ashok Nagar

Mayur Vihar ExtensionMayur

Vihar-I Akshardham

Yamuna Bank

Indraprastha Pragati

Maidan Mandi House

Barakhamba Road

Rajiv Chowk

R K Ashram Marg

Jhandewalan Karol

Bagh Rajendra Place

Patel Nagar Shadipur

Kirti Nagar Moti

Nagar Ramesh

Nagar Rajouri

GardenTagore

GardenSubhash

NagarJanakpuri

East Janakpuri

West

Uttam Nagar East

Uttam Nagar West

Nawada

Dwarka Mor

Dwarka

Dwarka Sector 14

Dwarka Sector 13

Dwarka Sector 12

Dwarka Sector 11

Dwarka Sector 10

Dwarka Sector 9

Dwarka Sector 8

Dwarka Sector 21

yellowline.txt

Jahangirpuri Adarsh

Nagar Azadpur

Model Town GTB

Nagar Vishwa

VidyalayaVidhan

Sabha Civil Lines

Kashmere Gate

Chandni Chowk

Chawri Bazar New

Delhi

Rajiv Chowk Patel

Chowk Central

SecretariatUdyog

Bhawan Race Course

Jor Bagh

INA

AIIMS

Green Park Hauz

Khas Malviya

NagarSaket

Qutub Minar

Chhatarpur

Sultanpur

Ghitorni Arjan

Garh

Guru Dronacharya

Sikandarpur

MG Road IFFCO

Chowk

HUDA City Centre

redline.txt Dilshad

Garden Jhilmil

Mansarovar Park Delhi

Shahdara Welcome

Seelampur

Shastri Park

Kashmere GateTis

Hazari

Pul   Bangash

Pratap   Nagar

Shastri Nagar

Inderlok Kanhiya

NagarKeshav

Puram

Netaji Subhash Place

Kohat Enclave Pitam

Pura

Rohini East Rohini

West Rithala

greenline.txt Kirti

Nagar

Satguru Ramsingh MargAshok

Park Main Punjabi Bagh East

Shivaji Park

Madipur

Paschim Vihar East

Paschim Vihar WestPeera

Garhi

Udyog Nagar

Surajmal Stadium

Nangloi

Nangloi Railway station

Rajdhani Park

Mundka **violetline.txt**

Central Secretariat

Khan Market

JLN Stadium

Jangpura Lajpat

Nagar Moolchand

Kailash Colony

Nehru Place

Kalkaji Mandir

Govind Puri Okhla

Jasola Apollo Sarita

Vihar Mohan Estate

Tughlakabad Badarpur

bluext.txt Yamuna

Bank Laxmi Nagar

Nirman Vihar Preet

Vihar Karkarduma

Anand Vihar ISBT

Kaushambi Vaishali

orangeline.txt New

Delhi

Shivaji Stadium

Dhaula Kuan Delhi

Aerocity Airport

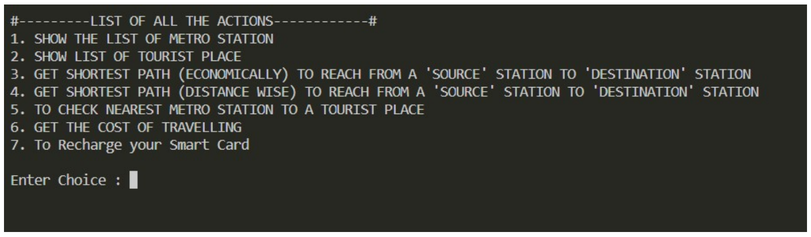
Dwarka Sector 21

```
#include<bits/stdc++.h>
#include<fstream>
#define ll long long
#define pb push_back
#define fi first
#define se second
#define mp make_pair
using namespace std;

map<string,ll>M;
// city , key(weight)
char color[200][200]={'\0'};
class comparedis
{
    public:
    bool operator()(pair<ll,ll> &p,pair<ll,ll> &q)
        {
            return (p.se > q.se); // For min heap use > sign
        }
};
vector< pair<ll,ll> > v[100010];//Adjacency matrix
ll N;// N is no of vertices
string station[200];
map <string,string> tourm;
void recharge()
{
    fstream f;
    ll amt,ini,cid,fin,x;
    ll c_id,amount;
    f.open("paisa.txt",ios::in|ios::out);
    if(!f)
        cout<<"Not Found\n"<<endl;
    f.seekg(0);
    cout<<endl;
    cout<<"Enter Card Id : ";
    cin>>c_id;
    cout<<"\nEnter Amount : ";
    cin>>amount;
    f.clear();
    while(!f.eof())
    {
        ini=f.tellg();
        f.ignore();
        f>>cid;
        f>>amt;
        fin=f.tellg();
        if(cid==c_id)
        {
            x=amt+amount;
            f.seekg(ini);
            f<<endl<<cid<<endl<<x;
            cout<<"Recharge Details\n";
            cout<<"\nCard Id: "<<cid<<endl;
            cout<<"Initial Balance: "<<amt<<endl;
            cout<<"Recharge Amount: "<<amount<<endl;
            cout<<"Total Balance: "<<x<<endl;
            break;
        }
    }
    f.close();
}
void gettour()
{
    ifstream fin;
    string s1,s2;
    fin.open("tourplace.txt",ios::in);
    if(!fin)
        cout<<"Not Found\n";
    fin.seekg(0);
    fin.clear();
    while(!fin.eof())
    {
        getline(fin,s1);
        getline(fin,s2);
        tourm[s1]=s2;
        //cout<<tourm[s1]<<endl;
    }
    fin.close();
    // map<string,string>:: iterator it;
    // for(it=tourm.begin();it!=tourm.end();it++){
    //  cout<<it->fi<<"-> "<<it->se<<endl;
    // }
}
//Given below code will print the path
void disp(ll src,ll dest,ll par[])
{
    ll i,x,y,cn=0,ci=0;
    stack<ll> st;
    st.push(dest);
    i=dest;
    while(par[i]!=-1)
    {
        i=par[i];
        st.push(i);
    }
    char col='\0';
    while(!st.empty())
    {
        x=st.top();
        st.pop();
        if(!st.empty())
            y=st.top();
        cout<<station[x]<<"-> ";
        cn++;
        if(col=='\0')
            col=color[x][y];
        else if(col!='\0'&&col!=color[x][y])
        {
            char c=color[x][y];
            ci++;
            if(c=='b')
                cout<<"\t\tChange to blue line";
            else if(c=='y')
                cout<<"\t\tChange to yellow line";
            else if(c=='o')
                cout<<"\t\tChange to orange line";
            else if(c=='g')
                cout<<"\t\tChange to green line";
            else if(c=='r')
                cout<<"\t\tChange to red line";
            else if(c=='v')
                cout<<"\t\tChange to Violet line";
            col=c;
        }
        cout<<endl;
    }
    // cout<<endl<<"No of stations ="<<cn<<endl;
    // cout<<"No of interchange stations ="<<ci-1<<endl;
    cout<<endl;
}
int cost(ll src,ll dest,ll par[])
{
    ll i,x,y,cn=0,ci=0;
    stack<ll> st;
    st.push(dest);
    i=dest;
    while(par[i]!=-1)
    {
        i=par[i];
        st.push(i);
    }
    char col='\0';
    while(!st.empty())
    {
        x=st.top();
        st.pop();
        if(!st.empty())
            y=st.top();
        cn++;
        if(col=='\0')
            col=color[x][y];
        else if(col!='\0'&&col!=color[x][y])
        {
            char c=color[x][y];
            ci++;
            col=c;
        }
        // cout<<endl;
    }
    int price;
    if(cn>0 && cn<10){
        price=10+6*(cn-1);
    }
    else if(cn>=10 && cn< 20){
        price=10+5*(cn-1);
    }
    else if(cn>=20){
        price=10+4*(cn-1);
    }
    return price;
}
//To find shotest path
void bfs(ll src,ll dest)
{
    bool vis[100010]={false};
    ll par[100010];
    for(ll i=0;i<N;i++)
        par[i]=-1;
    queue<ll> q;
    q.push(src);
    vis[src]=true;
    while(!q.empty())
    {
        ll x=q.front();
        q.pop();
        ll vsz=v[x].size();
        for(ll i=0;i<vsz;i++)
        {
            ll y=v[x][i].fi;
            if(!vis[y])
            {
                par[y]=x;
                vis[y]=true;
                q.push(y);
            }
        }
        v[x].clear();
    }
    disp(src,dest,par);
}
//To find most economical path
int dijkstra(ll src,ll dest,int d)
{
    bool vis[100010]={false};
    ll dist[100010], par[100010];
    for(ll i=0;i<N;i++)
    {
        dist[i]=LLONG_MAX;
        par[i]=-1;
    }
    priority_queue< pair<ll,ll>,vector< pair<ll,ll> >,comparedis > pq;
    pq.push(mp(src,0));
    dist[src]=0;
    par[src]=-1;
    vis[src]=true;
    while(!pq.empty())
    {
        pair<ll,ll> k=pq.top();
        pq.pop();
        ll x=k.fi;
        //if(x==dest)
        //  break;
        ll vsz=v[x].size();
        for(ll i=0;i<vsz;i++)
        {
            ll y=v[x][i].fi;
            ll w=v[x][i].se;
            if(dist[x]+w < dist[y])
            {
                par[y]=x;
                dist[y]=dist[x]+w;
            }
            if(!vis[y])
            {
                vis[y]=true;
                pq.push(mp(y,dist[y]));
            }
        }
        v[x].clear();
    }
    disp(src,dest,par);
    if(d==0){
        return 0;
    }
    else{
        int p = cost(src,dest,par);
        return p;
    }
}
void consmap()//To assign values to metro stations
{
    ifstream fin;
    string s;
    fin.open("list.txt",ios::in);
    ll l=0;
    fin.seekg(0);
    fin.clear();
    while(!fin.eof())
    {
        getline(fin,s);
        M[s]=l;
        station[l]=s;
        l++;
    }
    N=l-1;
    fin.close();
    map<string,ll> ::iterator it;
    //for(it=M.begin();it!=M.end();it++)
    //  cout<<it->se<<" "<<it->fi<<endl;
}
void addedge(char fname[],ll w)//To add edges
{
    ifstream fin;
    string s;
    ll x,y;
    fin.open(fname,ios::in);
    fin.seekg(0);
    getline(fin,s);
    x=M[s];
    char c=fname[0];
    fin.clear();
    while(!fin.eof())
    {
        getline(fin,s);
        y=M[s];
        v[x].pb(mp(y,w));
        v[y].pb(mp(x,w));
        color[x][y]=c;
        color[y][x]=c;
        x=y;
    }
    fin.close();
}
void consgraph()//To construct edges
{
    //string s;
    addedge("blueline.txt",0);
    addedge("yellowline.txt",0);
    addedge("redline.txt",0);
    addedge("greenline.txt",0);
    addedge("violetline.txt",0);
    addedge("bluext.txt",0);
    addedge("orangeline.txt",1);
}
int main()
{
    string source,destination;
    ll i,x,y,w,src,dest,k,choice,dec;
    char ch;
    gettour();
    consmap();
    do
    {
        system("cls");
        cout<<endl;
        cout<<"#---------LIST OF ALL THE ACTIONS------------#"<<endl;
        cout<<"1. SHOW THE LIST OF METRO STATION\n";
        cout<<"2. SHOW LIST OF TOURIST PLACE\n";
        cout<<"3. GET SHORTEST PATH (ECONOMICALLY) TO REACH FROM A 'SOURCE'
STATION TO 'DESTINATION' STATION\n";
        cout<<"4. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE'
STATION TO 'DESTINATION' STATION\n";
        cout<<"5. TO CHECK NEAREST METRO STATION TO A TOURIST PLACE\n";
        cout<<"6. GET THE COST OF TRAVELLING\n";
        cout<<"7. To Recharge your Smart Card\n";
        cout<<"8. Exit\n";
        cout<<"\nEnter Choice : ";
        cin>>dec;
        string s;
        ifstream f1,f2,f3,f4,f5,f6,f7,f8;
        switch(dec)
        {
            case 1:
                do
                    {
                        cout<<"\n#----------CHOOSE COLOUR OF LINE FOR METRO STATION----------
#\n";
                        cout<<"1. BlueLine Metro Stations\n";
                        cout<<"2. RedLine Metro Stations\n";
                        cout<<"3. GreenLine Metro Stations\n";
                        cout<<"4. VioletLine Metro Stations\n";
                        cout<<"5. YellowLine Metro Stations\n";
                        cout<<"6. BlueExt Metro Stations\n";
                        cout<<"7. OrangeLine Metro Stations\n";
                        cout<<"\nEnter Choice : ";
                        int cl;
                        cin>>cl;
                        switch(cl){
                            case 1:
                            f1.open("blueline.txt",ios::in);
                            while(true){
                                getline(f1,s);
                                if(f1.eof())
                                    break;
                                else {
                                    cout<<s<<"\n";
                                }
                            }
                            f1.close();
                            break;
                            case 2:
                            f2.open("redline.txt",ios::in);
                            while(true){
                                getline(f2,s);
    if(f2.eof())
        break;
    else {
        cout<<s<<"\n";
    }
}
f2.close();
break;
case 3:
f3.open("greenline.txt",ios::in);
while(true){
    getline(f3,s);
    if(f3.eof())
        break;
    else {
        cout<<s<<"\n";
    }
}
f3.close();
break;
case 4:
f4.open("violetline.txt",ios::in);
while(true){
    getline(f4,s);
    if(f4.eof())
        break;
    else {
        cout<<s<<"\n";
    }
}
f4.close();
break;
case 5:
f5.open("yellowline.txt",ios::in);
while(true){
    getline(f5,s);
    if(f5.eof())
        break;
    else {
        cout<<s<<"\n";
    }
}
f5.close();
break;
case 6:
f6.open("bluext.txt",ios::in);
while(true){
    getline(f6,s);
    if(f6.eof())
        break;
    else {
        cout<<s<<"\n";
```

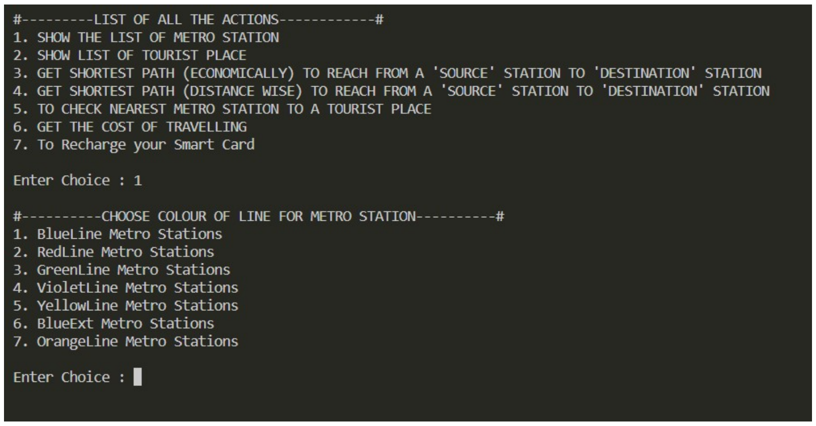
```
                    }
                }
                f6.close();
                break;
                case 7:
                f7.open("orangeline.txt",ios::in);
                while(true){
                    getline(f7,s);
                    if(f7.eof())
                        break;
                    else {
                        cout<<s<<"\n";
                    }
                }
                f7.close();
                break;
            }
            cout<<"\nDo you wish to check for any other list of station(Y/N) : ";
            cin>>ch;
        }while(ch=='Y'||ch=='y');
        break;
case 2:
        cout<<"\nList OF TOURIST PLACES\n\n";
            f8.open("tourplace.txt",ios::in);
                while(true){
                    getline(f8,s);
                    if(f8.eof())
                        break;
                    else {
                        cout<<s<<"\n";
                    }
                }
                f8.close();
            break;
case 3:
        do
        {
            consgraph();//To build the adjacency matrix
            cout<<"\nEnter station 1 : ";
            //getline(cin,source);
            fflush(stdin);
            getline(cin,source);
            //cout<<source<<endl;
            cout<<"\nEnter station 2 : ";
            getline(cin,destination);
            //cout<<destination<<endl;
            src=M[source];
            dest=M[destination];
            bfs(src,dest);
            cout<<"Do you wish to check for any other station(Y/N) : ";
            cin>>ch;
        }while(ch=='Y'||ch=='y');
        break;
case 4:
        do
        {
            consgraph();
            cout<<"\nEnter station 1 : ";
            fflush(stdin);
            getline(cin,source);
            cout<<"\nEnter station 2 : ";
            getline(cin,destination);
            src=M[source];
            dest=M[destination];
            dijkstra(src,dest,0);
            cout<<"Do you wish to check for any other station(Y/N) : ";
            cin>>ch;
        }while(ch=='Y'||ch=='y');
        break;
case 5:
        do
        {
            string place;
            cout<<"\nEnter a place : ";
            fflush(stdin);
            //getline(cin,place);
            getline(cin,place);
            string st;
            st=tourm[place];
            cout<<st<<endl;
            cout<<"\nDo you wish to check for any other place(Y/N) : ";
            cin>>ch;
        }while(ch=='Y'||ch=='y');
        break;
case 6:
        do
        {
            consgraph();
            cout<<"\nEnter station 1 : ";
            fflush(stdin);
            getline(cin,source);
            cout<<"\nEnter station 2 : ";
            getline(cin,destination);
            src=M[source];
            dest=M[destination];
            int ans=dijkstra(src,dest,1);
            cout<<"Total price: "<<ans<<endl;
            cout<<"\nDo you wish to check for another path(Y/N) : ";
            cin>>ch;
        }while(ch=='Y'||ch=='y');
        break;
case 7:
        do
        {
```

```
                    recharge();
                    cout<<"\nDo you wish to recharge some other smart card(Y/N) : ";
                    cin>>ch;
                }while(ch=='Y'||ch=='y');
                break;
            case 8:
                exit(0);
        }
        cout<<"\nDo you wish to go back to main menu(Y/N) : ";
        cin>>ch;
    }while(ch=='Y'||ch=='y');
    return 0;
}
```

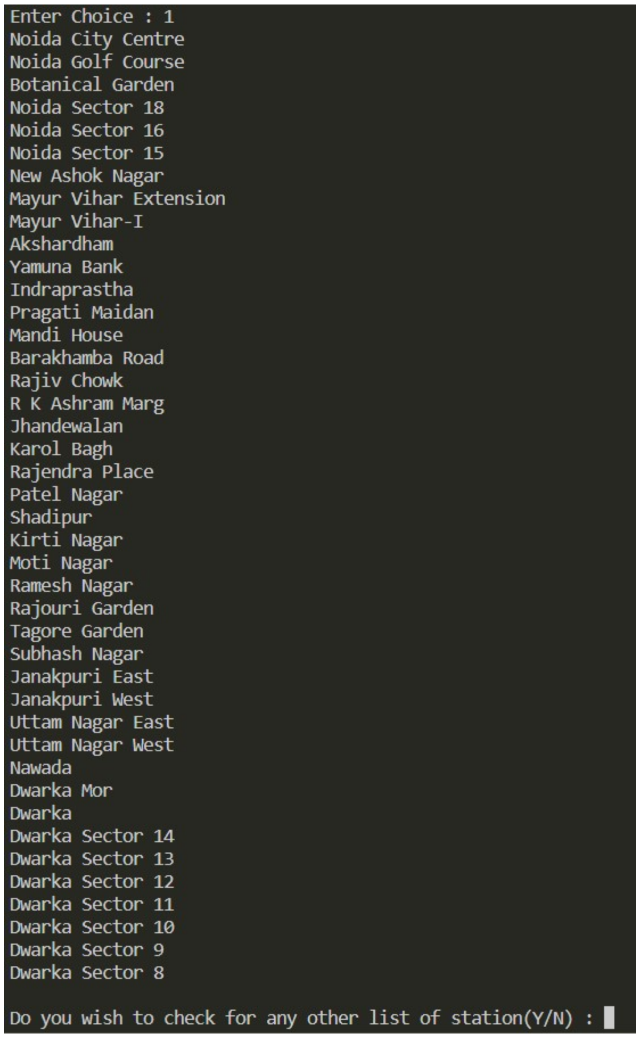
Output



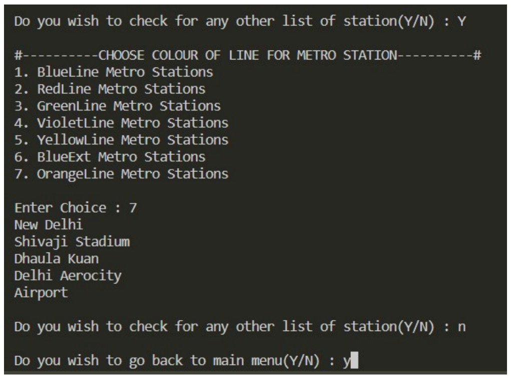
```
#---------LIST OF ALL THE ACTIONS-----------#
1. SHOW THE LIST OF METRO STATION
2. SHOW LIST OF TOURIST PLACE
3. GET SHORTEST PATH (ECONOMICALLY) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. TO CHECK NEAREST METRO STATION TO A TOURIST PLACE
6. GET THE COST OF TRAVELLING
7. To Recharge your Smart Card

Enter Choice :
```



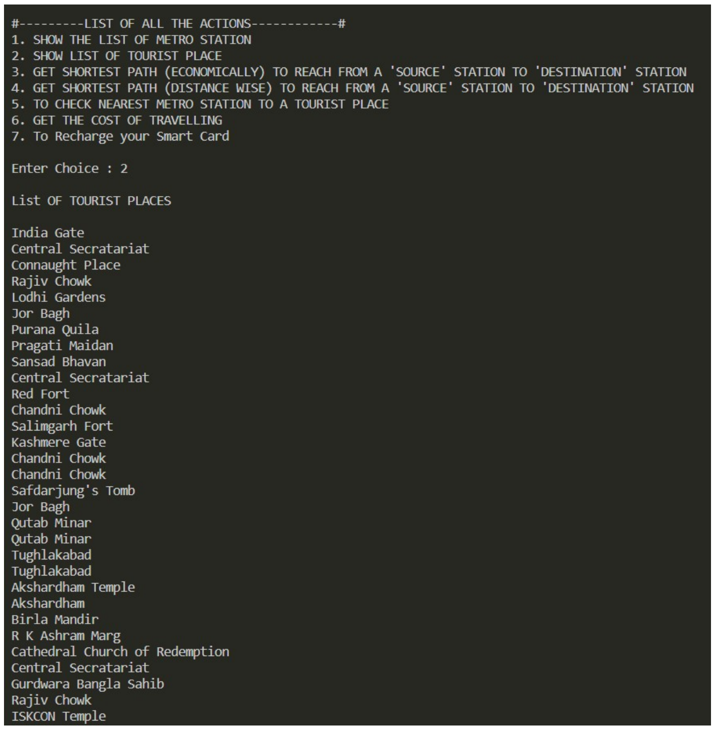
```
#---------LIST OF ALL THE ACTIONS-----------#
1. SHOW THE LIST OF METRO STATION
2. SHOW LIST OF TOURIST PLACE
3. GET SHORTEST PATH (ECONOMICALLY) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. TO CHECK NEAREST METRO STATION TO A TOURIST PLACE
6. GET THE COST OF TRAVELLING
7. To Recharge your Smart Card

Enter Choice : 1

#---------CHOOSE COLOUR OF LINE FOR METRO STATION---------#
1. BlueLine Metro Stations
2. RedLine Metro Stations
3. GreenLine Metro Stations
4. VioletLine Metro Stations
5. YellowLine Metro Stations
6. BlueExt Metro Stations
7. OrangeLine Metro Stations

Enter Choice :
```



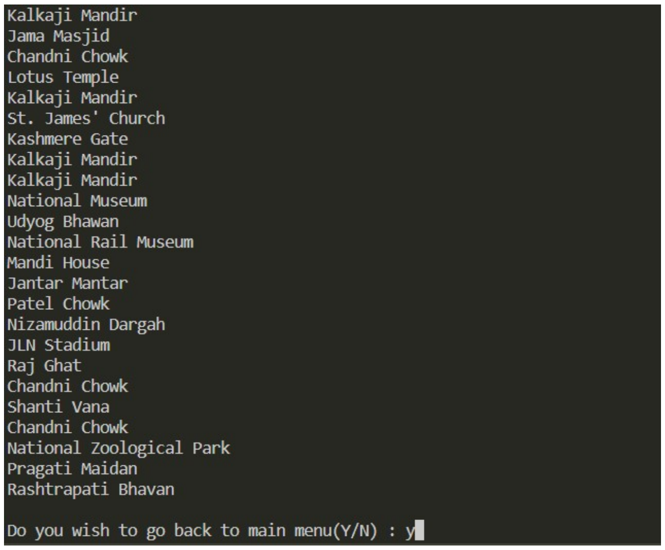
```
Enter Choice : 1
Noida City Centre
Noida Golf Course
Botanical Garden
Noida Sector 18
Noida Sector 16
Noida Sector 15
New Ashok Nagar
Mayur Vihar Extension
Mayur Vihar-I
Akshardham
Yamuna Bank
Indraprastha
Pragati Maidan
Mandi House
Barakhamba Road
Rajiv Chowk
R K Ashram Marg
Jhandewalan
Karol Bagh
Rajendra Place
Patel Nagar
Shadipur
Kirti Nagar
Moti Nagar
Ramesh Nagar
Rajouri Garden
Tagore Garden
Subhash Nagar
Janakpuri East
Janakpuri West
Uttam Nagar East
Uttam Nagar West
Nawada
Dwarka Mor
Dwarka
Dwarka Sector 14
Dwarka Sector 13
Dwarka Sector 12
Dwarka Sector 11
Dwarka Sector 10
Dwarka Sector 9
Dwarka Sector 8

Do you wish to check for any other list of station(Y/N) :
```



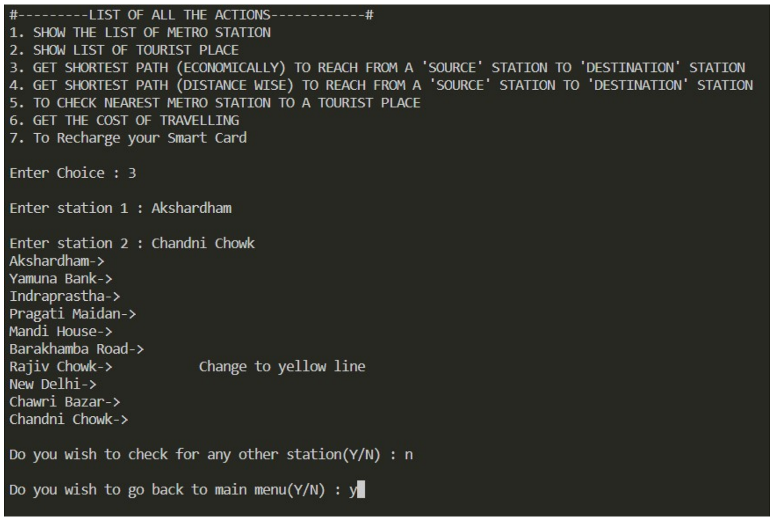
```
Do you wish to check for any other list of station(Y/N) : Y

#---------CHOOSE COLOUR OF LINE FOR METRO STATION---------#
1. BlueLine Metro Stations
2. RedLine Metro Stations
3. GreenLine Metro Stations
4. VioletLine Metro Stations
5. YellowLine Metro Stations
6. BlueExt Metro Stations
7. OrangeLine Metro Stations

Enter Choice : 7
New Delhi
Shivaji Stadium
Dhaula Kuan
Delhi Aerocity
Airport

Do you wish to check for any other list of station(Y/N) : n

Do you wish to go back to main menu(Y/N) : y
```



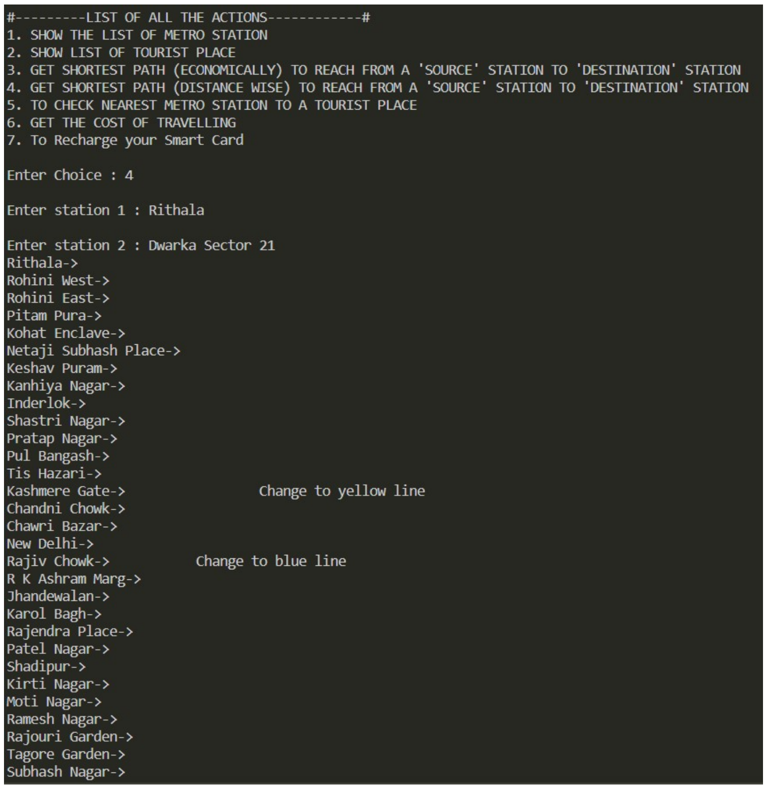
```
#---------LIST OF ALL THE ACTIONS-----------#
1. SHOW THE LIST OF METRO STATION
2. SHOW LIST OF TOURIST PLACE
3. GET SHORTEST PATH (ECONOMICALLY) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. TO CHECK NEAREST METRO STATION TO A TOURIST PLACE
6. GET THE COST OF TRAVELLING
7. To Recharge your Smart Card

Enter Choice : 2

List OF TOURIST PLACES

India Gate
Central Secratariat
Connaught Place
Rajiv Chowk
Lodhi Gardens
Jor Bagh
Purana Quila
Pragati Maidan
Sansad Bhavan
Central Secratariat
Red Fort
Chandni Chowk
Salimgarh Fort
Kashmere Gate
Chandni Chowk
Chandni Chowk
Safdarjung's Tomb
Jor Bagh
Qutab Minar
Qutab Minar
Tughlakabad
Tughlakabad
Akshardham Temple
Akshardham
Birla Mandir
R K Ashram Marg
Cathedral Church of Redemption
Central Secratariat
Gurdwara Bangla Sahib
Rajiv Chowk
ISKCON Temple
```



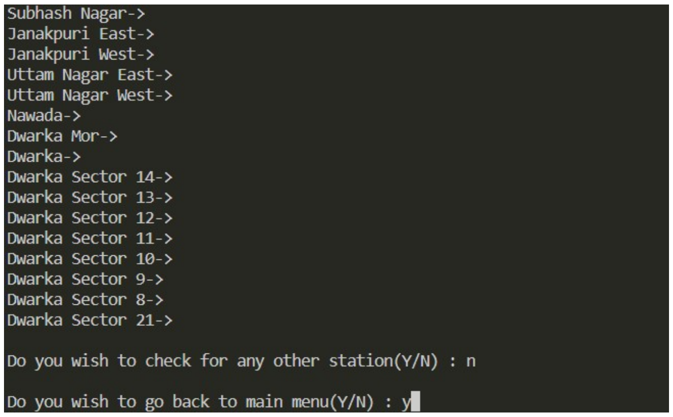
```
Kalkaji Mandir
Jama Masjid
Chandni Chowk
Lotus Temple
Kalkaji Mandir
St. James' Church
Kashmere Gate
Kalkaji Mandir
Kalkaji Mandir
National Museum
Udyog Bhawan
National Rail Museum
Mandi House
Jantar Mantar
Patel Chowk
Nizamuddin Dargah
JLN Stadium
Raj Ghat
Chandni Chowk
Shanti Vana
Chandni Chowk
National Zoological Park
Pragati Maidan
Rashtrapati Bhavan

Do you wish to go back to main menu(Y/N) : y
```



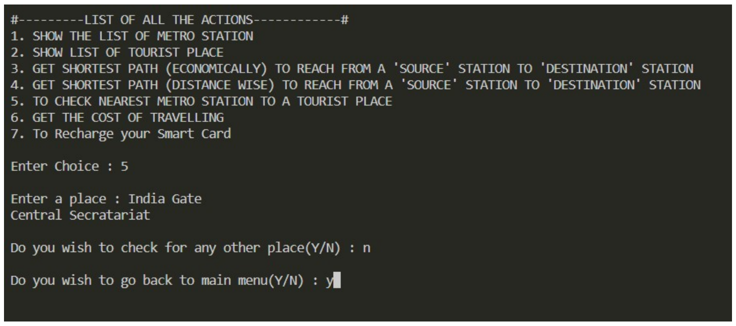
```
#---------LIST OF ALL THE ACTIONS------------#
1. SHOW THE LIST OF METRO STATION
2. SHOW LIST OF TOURIST PLACE
3. GET SHORTEST PATH (ECONOMICALLY) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. TO CHECK NEAREST METRO STATION TO A TOURIST PLACE
6. GET THE COST OF TRAVELLING
7. To Recharge your Smart Card

Enter Choice : 3

Enter station 1 : Akshardham

Enter station 2 : Chandni Chowk
Akshardham->
Yamuna Bank->
Indraprastha->
Pragati Maidan->
Mandi House->
Barakhamba Road->
Rajiv Chowk->           Change to yellow line
New Delhi->
Chawri Bazar->
Chandni Chowk->

Do you wish to check for any other station(Y/N) : n

Do you wish to go back to main menu(Y/N) : y
```

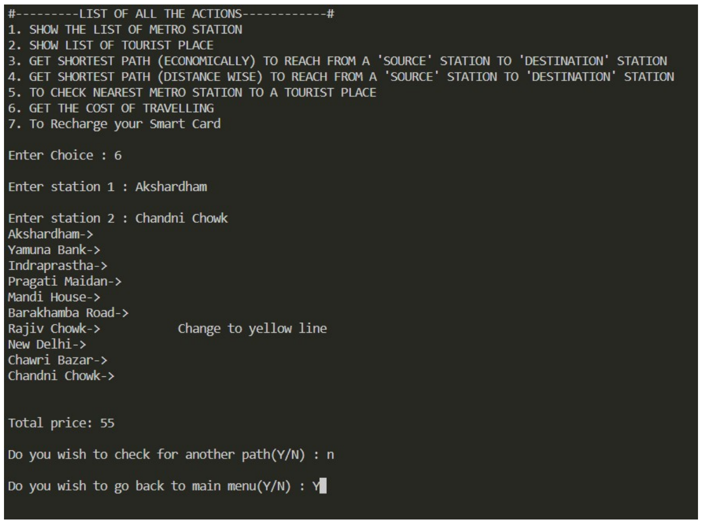


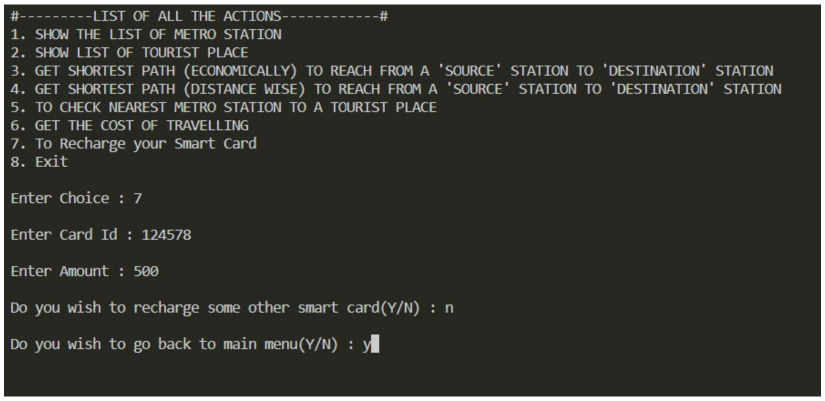
```
#---------LIST OF ALL THE ACTIONS------------#
1. SHOW THE LIST OF METRO STATION
2. SHOW LIST OF TOURIST PLACE
3. GET SHORTEST PATH (ECONOMICALLY) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. TO CHECK NEAREST METRO STATION TO A TOURIST PLACE
6. GET THE COST OF TRAVELLING
7. To Recharge your Smart Card

Enter Choice : 4

Enter station 1 : Rithala

Enter station 2 : Dwarka Sector 21
Rithala->
Rohini West->
Rohini East->
Pitam Pura->
Kohat Enclave->
Netaji Subhash Place->
Keshav Puram->
Kanhiya Nagar->
Inderlok->
Shastri Nagar->
Pratap Nagar->
Pul Bangash->
Tis Hazari->
Kashmere Gate->                 Change to yellow line
Chandni Chowk->
Chawri Bazar->
New Delhi->
Rajiv Chowk->           Change to blue line
R K Ashram Marg->
Jhandewalan->
Karol Bagh->
Rajendra Place->
Patel Nagar->
Shadipur->
Kirti Nagar->
Moti Nagar->
Ramesh Nagar->
Rajouri Garden->
Tagore Garden->
Subhash Nagar->
```

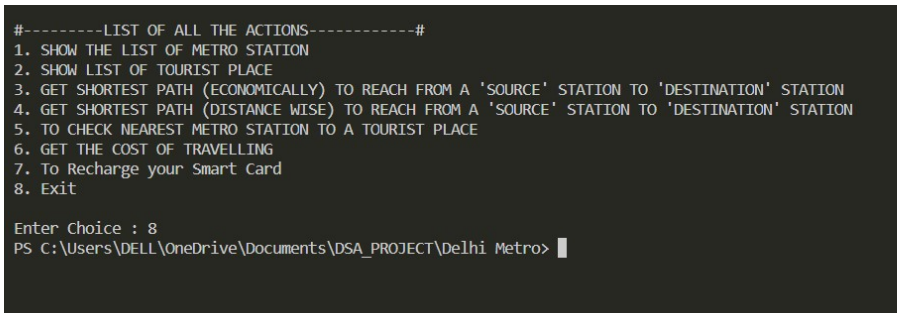


```
Subhash Nagar->
Janakpuri East->
Janakpuri West->
Uttam Nagar East->
Uttam Nagar West->
Nawada->
Dwarka Mor->
Dwarka->
Dwarka Sector 14->
Dwarka Sector 13->
Dwarka Sector 12->
Dwarka Sector 11->
Dwarka Sector 10->
Dwarka Sector 9->
Dwarka Sector 8->
Dwarka Sector 21->

Do you wish to check for any other station(Y/N) : n

Do you wish to go back to main menu(Y/N) : y
```


```
#---------LIST OF ALL THE ACTIONS------------#
1. SHOW THE LIST OF METRO STATION
2. SHOW LIST OF TOURIST PLACE
3. GET SHORTEST PATH (ECONOMICALLY) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. TO CHECK NEAREST METRO STATION TO A TOURIST PLACE
6. GET THE COST OF TRAVELLING
7. To Recharge your Smart Card

Enter Choice : 5

Enter a place : India Gate
Central Secratariat

Do you wish to check for any other place(Y/N) : n

Do you wish to go back to main menu(Y/N) : y
```


```
#---------LIST OF ALL THE ACTIONS------------#
1. SHOW THE LIST OF METRO STATION
2. SHOW LIST OF TOURIST PLACE
3. GET SHORTEST PATH (ECONOMICALLY) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. TO CHECK NEAREST METRO STATION TO A TOURIST PLACE
6. GET THE COST OF TRAVELLING
7. To Recharge your Smart Card

Enter Choice : 6

Enter station 1 : Akshardham

Enter station 2 : Chandni Chowk
Akshardham->
Yamuna Bank->
Indraprastha->
Pragati Maidan->
Mandi House->
Barakhamba Road->
Rajiv Chowk->           Change to yellow line
New Delhi->
Chawri Bazar->
Chandni Chowk->


Total price: 55

Do you wish to check for another path(Y/N) : n

Do you wish to go back to main menu(Y/N) : Y
```


```
#---------LIST OF ALL THE ACTIONS------------#
1. SHOW THE LIST OF METRO STATION
2. SHOW LIST OF TOURIST PLACE
3. GET SHORTEST PATH (ECONOMICALLY) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. TO CHECK NEAREST METRO STATION TO A TOURIST PLACE
6. GET THE COST OF TRAVELLING
7. To Recharge your Smart Card
8. Exit

Enter Choice : 7

Enter Card Id : 124578

Enter Amount : 500

Do you wish to recharge some other smart card(Y/N) : n

Do you wish to go back to main menu(Y/N) : y
```


```
#---------LIST OF ALL THE ACTIONS------------#
1. SHOW THE LIST OF METRO STATION
2. SHOW LIST OF TOURIST PLACE
3. GET SHORTEST PATH (ECONOMICALLY) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. TO CHECK NEAREST METRO STATION TO A TOURIST PLACE
6. GET THE COST OF TRAVELLING
7. To Recharge your Smart Card
8. Exit

Enter Choice : 8
PS C:\Users\DELL\OneDrive\Documents\DSA_PROJECT\Delhi Metro>
```

Conclusion and Future

Built a project using C++, which finds shortest and most economical path between two travel destinations on Delhi metro. Dijkstra's algorithm was used to find the shortest path and economical was found using BFS of the metro map.

It also finds nearest metro station to popular tourist destinations likeIndia Gate.

There is also a scope for future metro lines that can be interconnectedwith other existing metro lines and this project is also ready for further infrastructure developments.

References

- https://www.geeksforgeeks.org/
- https://cplusplus.com/
- https://www.tutorialspoint.com/index.htm
- https://www.youtube.com/