DSA PROJECT

SUBJECT :-

DATA STRUCTURE

TOPIC :-

RAILWAY INDICATOR

Introduction

In this project, **2 way circular linked list** data structure is used. Linked list represents the stations with their next station and previous station. Each node contains the name of the station, time, next station pointer, previous station pointer, distance till next station and index.

The code contains 2 pre-defined two way circular linked lists. First list’s head is Panvel station. Second list’s head is Thane station. Each Station is identified by its unique number called index. On the basis of index, we are going to do the operations on those lists. Even if the same stations are from different lists, the indexes are same. For eg. Turbhe station is in both the lists but the indes in both the lists for Turbhe station is same.

There are 8 important functions in the program.

1. InsertAtHead()

Inserts the new record at head.

2. InsertAtTail()

Inserts the new record at end.

3. makeCycle()

Makes cycle at the end by pointing end->next to head and head->prev to end.

4. takeInput()

Takes input of starting station and ending station.

5. getTime()

calculates the time to reach the ending station form the given station.

6. getRoute()

Finds the best route to reach the destination station from starting station.

7. getDistance()

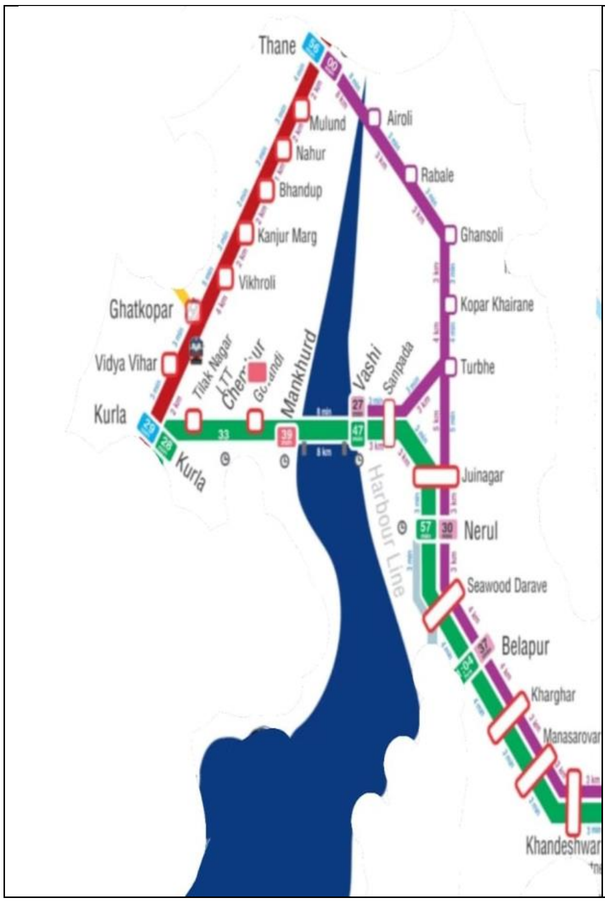
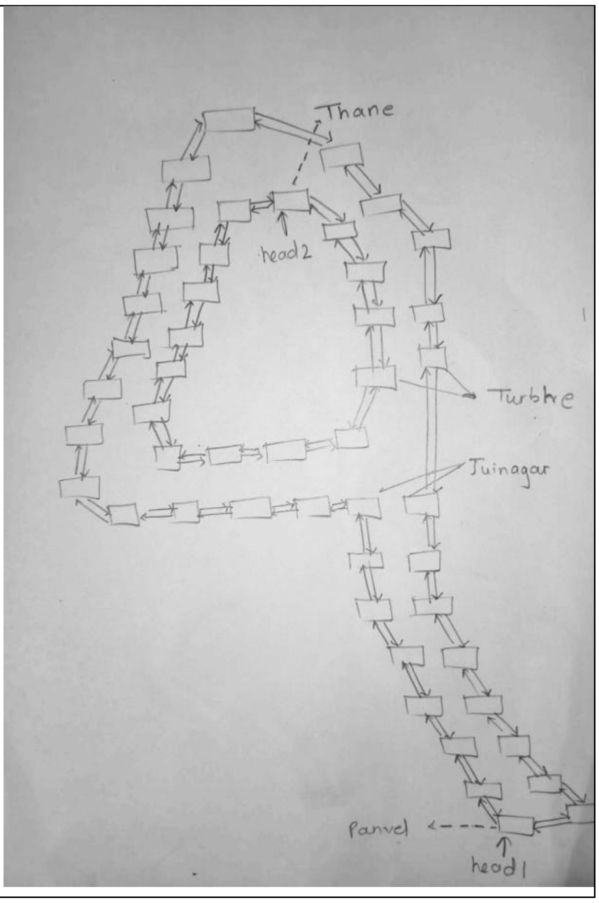
Calculates the distance to reach the ending station form the given station.

8. getCount()

Counts the stations between starting and ending stations.

In the main menu of the program, there are five options. Find best route, find distance, find travelling time, count stations and exit. All of these options except exit will take input of starting station and ending station. According to input, it will show the result.

For visualization purpose, I am showing the following map


Thane
Mulund
Nahur
Bhandup
Kanjur Marg
Vikhroli
Ghatkopar
Vidya Vihar
Kurla
Tilak Nagar
LTT
Chembur
Govandi
Mankhurd
Vashi
Sanpada
Airoli
Rabale
Ghansoli
Kopar Khairane
Turbhe
Juinagar
Nerul
Harbour Line
Seawood Darave
Belapur
Kharghar
Manasarovar
Khandeshwar

Thane
head2
Turbhe
Juinagar
Panvel
head1

Source Code:

```
#include <iostream>
using namespace std;

class node
{
public:
    string data;
    int index, dist, time;
    node *next;
    node *prev;
    node(string val, int no, int distance, int timetaken)
    {
        data = val;
        index = no;
        dist = distance;
        time = timetaken;
        next = NULL;
        prev = NULL;
    }
};
struct returnVal
{
    node *start;
    node *end;
    char rotate;
} returnValues;

int station1, station2;
node *head1, *head2, *startSt, *endSt;
```

```
void insertAtHead(node *&head, string val, int no, int distance, int timetaken)
{
    node *n = new node(val, no, distance, timetaken);
    n->next = head;
    if (head != NULL)
    {
        head->prev = n;
    }
    head = n;
}
void insertAtTail(node *&head, string val, int no, int distance, int timetaken)
{
    if (head == NULL)
    {
        insertAtHead(head, val, no, distance, timetaken);
        return;
    }
    node *n = new node(val, no, distance, timetaken);
    node *temp = head;
    while (temp->next != NULL)
    {
        temp = temp->next;
    }
    temp->next = n;
    n->prev = temp;
}

void makecycle(node *&head)
{
    node *temp = head;
    while (temp->next != NULL)
    {
        temp = temp->next;
    }
    temp->next = head;
    head->prev = temp;
}

void takeInput()
{

    cout << "\n\n\t1.Panvel \t2.Khandeshwar  \t3.Mansarovar    \t4.Kharghar\n";
    cout << "\t5.Belapur    \t6.Seawoods     \t7.Nerul         \t8.Juinagar \n";
    cout << "\t9.Sanpada    \t10.Vashi       \t11.Mankhurd     \t12.Govandi \n";
    cout << "\t13.Chembur   \t14.Tilak Nagar \t15.Kurla        \t16.Vidya Vihar \n";
    cout << "\t17.Ghatkopar \t18.Vikhroli    \t19.Kanjur Marg  \t20.Bhandup \n";
    cout << "\t21.Nahur     \t22.Mulund      \t23.Thane        \t24.Arioli \n";
    cout << "\t25.Rabale    \t26.Ghansoli    \t27.Koparkhairane\t28.Turbhe \n";

input1:
    cout << "Enter starting station :";
    cin >> station1;
    if (station1 > 28 || station1 < 1)
    {
        cout << "Enter Valid Input!!\n";
        goto input1;
    }
input2:
    cout << "Enter ending Station :";
    cin >> station2;
    if (station2 > 28 || station2 < 1)
    {
        cout << "Enter Valid Input!!\n";
        goto input2;
```

```
}
node *temp;
if ((station1 > 8 && station1 < 29) && (station2 > 8 && station2 < 29))
{
    temp = head1;
    while (temp->index != station1)
    {
        temp = temp->next;
    }
    startSt = temp;
    temp = head1;
    while (temp->index != station2)
    {
        temp = temp->next;
    }
    endSt = temp;
}
else
{
    temp = head2;
    while (temp->index != station1)
    {
        temp = temp->next;
    }
    startSt = temp;
    temp = head2;
    while (temp->index != station2)
    {
        temp = temp->next;
    }
    endSt = temp;
}
```

```
}

int getTime(node *st1, node *st2, char direction)
{
    node *temp = st1;
    int time = 0;

    if (direction == 'C')
    {
        while (temp->index != st2->index)
        {
            time += temp->time;
            temp = temp->next;
        }
    }
    else
    {
        while (temp->index != st2->index)
        {
            time += temp->prev->time;
            temp = temp->prev;
        }
    }
    return time;
}

returnVal getRoute(node *st1, node *st2)
{
    int t1, t2;
    node *temp;
    t1 = getTime(st1, st2, 'C');
```

```
if ((st1->index > 1 && st1->index <= 8) && (st2->index > 8))
{
    temp = st1->prev;
    while (temp->index != st1->index)
    {
        temp = temp->prev;
    }
    t2 = getTime(temp, st2, 'A');
    if (t1 > t2)
    {
        returnValues.start = temp;
        returnValues.end = st2;
        returnValues.rotate = 'A';
    }
    else
    {
        returnValues.start = st1;
        returnValues.end = st2;
        returnValues.rotate = 'C';
    }
}
else
{
    t2 = getTime(st1, st2, 'A');
    if (t2 < t1)
    {
        returnValues.start = st1;
        returnValues.end = st2;
        returnValues.rotate = 'A';
    }
    else
    {
            returnValues.start = st1;
            returnValues.end = st2;
            returnValues.rotate = 'C';
        }
    }
    return returnValues;
}

int getDistance(node *st1, node *st2, char direction)
{
    node *temp = st1;
    int dst = 0;

    if (direction == 'C')
    {
        while (temp->index != st2->index)
        {
            dst += temp->dist;
            temp = temp->next;
        }
    }
    else
    {
        while (temp->index != st2->index)
        {
            dst += temp->prev->dist;
            temp = temp->prev;
        }
    }
    return dst;
}
```

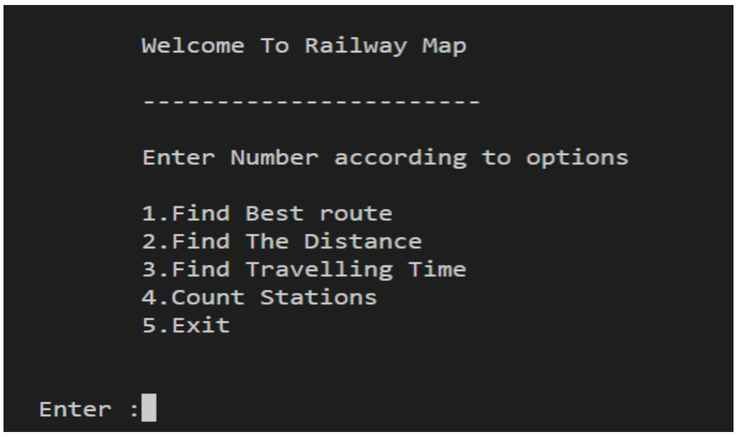
```
int getCount(node *st1, node *st2, char direction)
{
    node *temp = st1;
    int count = 0;

    if (direction == 'C')
    {
        while (temp->index != st2->index)
        {
            count++;
            temp = temp->next;
        }
    }
    else
    {
        while (temp->index != st2->index)
        {
            count++;
            temp = temp->prev;
        }
    }
    return count;
}

void findRoute()
{
    takeInput();
    returnVal values = getRoute(startSt, endSt);
    node *temp = values.start;
    cout << "Best route for " << startSt->data << " To " << endSt->data << " :" << endl;
    if (values.rotate == 'C')
    {
        while (temp != values.end)
        {
            cout << temp->data << "->";
            temp = temp->next;
        }
        cout << temp->data << endl;
    }
    else
    {
        while (temp != values.end)
        {
            cout << temp->data << "->";
            temp = temp->prev;
        }
        cout << temp->data << endl;
    }
}

void findDistance()
{
    takeInput();
    returnVal values = getRoute(startSt, endSt);
    int dist = getDistance(values.start, values.end, values.rotate);
    cout << "Distance from " <<startSt->data<<" to "<<endSt->data<<"  = " << dist << " km" << endl;
}
void findTime()
{
    takeInput();
    returnVal values = getRoute(startSt, endSt);
    int time = getTime(values.start, values.end, values.rotate);
    cout << "Traveling time from "<<startSt->data<<" to "<<endSt->data<<"  = " << time <<"
minutes"<< endl;
}
```

```
void findCount()
{
    takeInput();
    returnVal values = getRoute(startSt, endSt);
    int count = getCount(values.start, values.end, values.rotate);
    cout << "Number of stations from "<<startSt->data<<" to "<<endSt->data<<"  = " << count <<
endl;
}
void menu()
{
    do
    {
        cout << "\n\n\tWelcome To Railway Map \n\n";
        cout << "\t-----------------------\n\n";
        cout << "\tEnter Number according to options\n\n";
        cout << "\t1.Find Best rout\n";
        cout << "\t2.Find The Distance\n";
        cout << "\t3.Find Travelling Time\n";
        cout << "\t4.Count Stations\n";
        cout << "\t5.Exit\n";
    options:
        cout << "\n\n Enter :";
        int input;
        cin >> input;
        switch (input)
        {
        case (1):
            findRoute();
            break;
        case (2):
            findDistance();
            break;
        case (3):
            findTime();
            break;
        case (4):
            findCount();
            break;
        case (5):
            cout << "\n\t\t\t\t\t\tThank You for choosing us !!!\n\n";
            exit(0);
        default:
            cout << "Invalid Input! Enter Valid input";
            goto options;
            break;
        }
    } while (true);
}
int main()
{
    insertAtTail(head1, "Thane", 23, 8, 8);
    insertAtTail(head1, "Airoli", 24, 3, 3);
    insertAtTail(head1, "Rabale", 25, 3, 3);
    insertAtTail(head1, "Ghansoli", 26, 3, 3);
    insertAtTail(head1, "Koparkhairane", 27, 4, 4);
    insertAtTail(head1, "Turbhe", 28, 3, 4);
    insertAtTail(head1, "Sanpada", 9, 3, 3);
    insertAtTail(head1, "Vashi", 10, 8, 8);
    insertAtTail(head1, "Mankhurd", 11, 3, 3);    //
    insertAtTail(head1, "Govandi", 12, 3, 2);     //
    insertAtTail(head1, "Chembur", 13, 3, 3);     //
    insertAtTail(head1, "Tilak Nagar", 14, 3, 3); //
    insertAtTail(head1, "Kurla", 15, 2, 3);
    insertAtTail(head1, "Vidya Vihar", 16, 2, 3);
    insertAtTail(head1, "Ghatkopar", 17, 4, 4);
    insertAtTail(head1, "Vikhroli", 18, 2, 3);
    insertAtTail(head1, "Kanjur Marg", 19, 2, 3);
    insertAtTail(head1, "Bhandup", 20, 1, 3);
    insertAtTail(head1, "Nahur", 21, 2, 3);
    insertAtTail(head1, "Mulund", 22, 2, 6);
    makecycle(head1);

    insertAtTail(head2, "Panvel", 1, 3, 5);
    insertAtTail(head2, "Khandeshwar", 2, 3, 3);
    insertAtTail(head2, "Mansarovar", 3, 3, 3);
    insertAtTail(head2, "Kharghar", 4, 4, 4);
    insertAtTail(head2, "CBD_Belapur", 5, 4, 4);
    insertAtTail(head2, "SeaWoods", 6, 3, 3);
    insertAtTail(head2, "Nerul", 7, 3, 3);
    insertAtTail(head2, "Juinagar", 8, 3, 3);
    insertAtTail(head2, "Sanpada", 9, 3, 2);
    insertAtTail(head2, "Vashi", 10, 8, 8);
    insertAtTail(head2, "Mankhurd", 11, 3, 3);    //
    insertAtTail(head2, "Govandi", 12, 3, 2);     //
    insertAtTail(head2, "Chembur", 13, 3, 3);     //
    insertAtTail(head2, "Tilak Nagar", 14, 3, 3); //
    insertAtTail(head2, "Kurla", 15, 2, 3);
    insertAtTail(head2, "Vidya Vihar", 16, 2, 3);
    insertAtTail(head2, "Ghatkopar", 17, 4, 4);
    insertAtTail(head2, "Vikhroli", 18, 2, 3);
    insertAtTail(head2, "Kanjur Marg", 19, 2, 3);
    insertAtTail(head2, "Bhandup", 20, 1, 3);
    insertAtTail(head2, "Nahur", 21, 2, 3);
    insertAtTail(head2, "Mulund", 22, 2, 6);
    insertAtTail(head2, "Thane", 23, 8, 8);
    insertAtTail(head2, "Airoli", 24, 3, 3);
    insertAtTail(head2, "Rabale", 25, 3, 3);
```

```
    insertAtTail(head2, "Ghansoli", 26, 3, 3);
    insertAtTail(head2, "Koparkhairane", 27, 4, 4);
    insertAtTail(head2, "Turbhe", 28, 5, 4);
    insertAtTail(head2, "Juinagar", 8, 3, 5);
    insertAtTail(head2, "Nerul", 7, 3, 4);
    insertAtTail(head2, "SeaWoods", 6, 4, 4);
    insertAtTail(head2, "CBD_Belapur", 5, 4, 4);
    insertAtTail(head2, "Kharghar", 4, 3, 3);
    insertAtTail(head2, "Mansarovar", 3, 3, 3);
    insertAtTail(head2, "Khandeshwar", 2, 3, 6);
    makecycle(head2);
    menu();
}
```

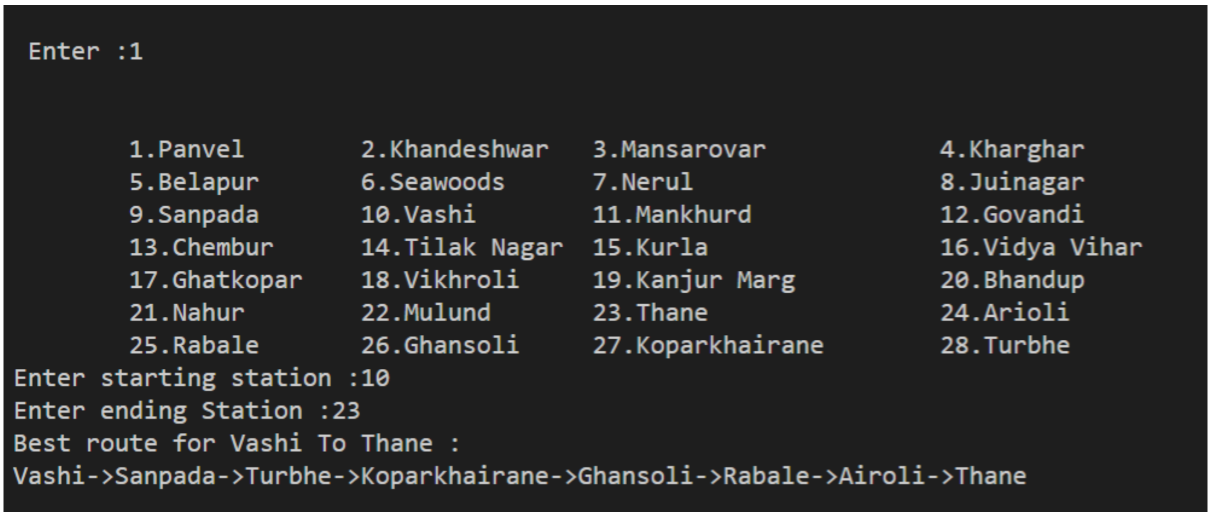
Output:

- Main Menu:



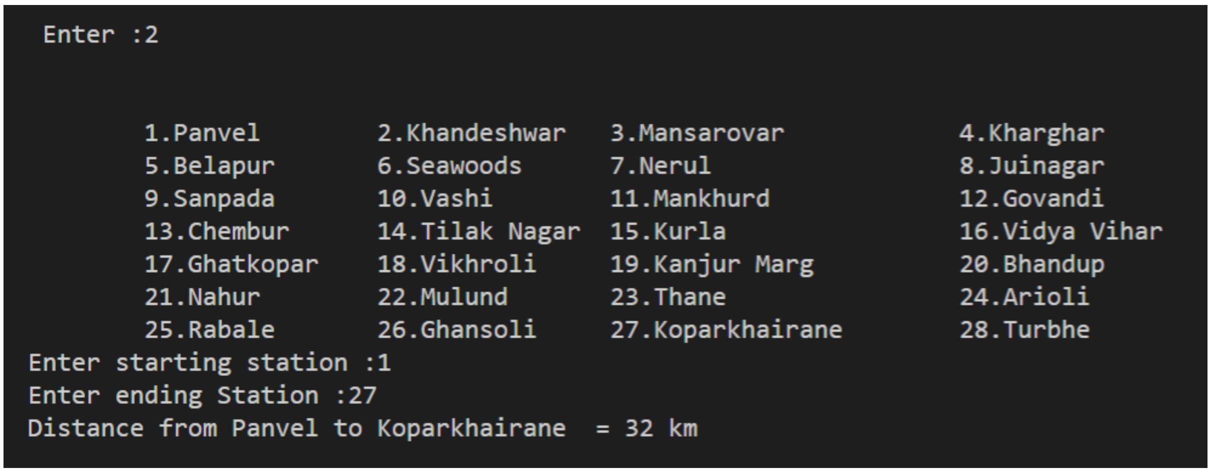
```
        Welcome To Railway Map

        -----------------------

        Enter Number according to options

        1.Find Best route
        2.Find The Distance
        3.Find Travelling Time
        4.Count Stations
        5.Exit


 Enter :
```

- Find best route



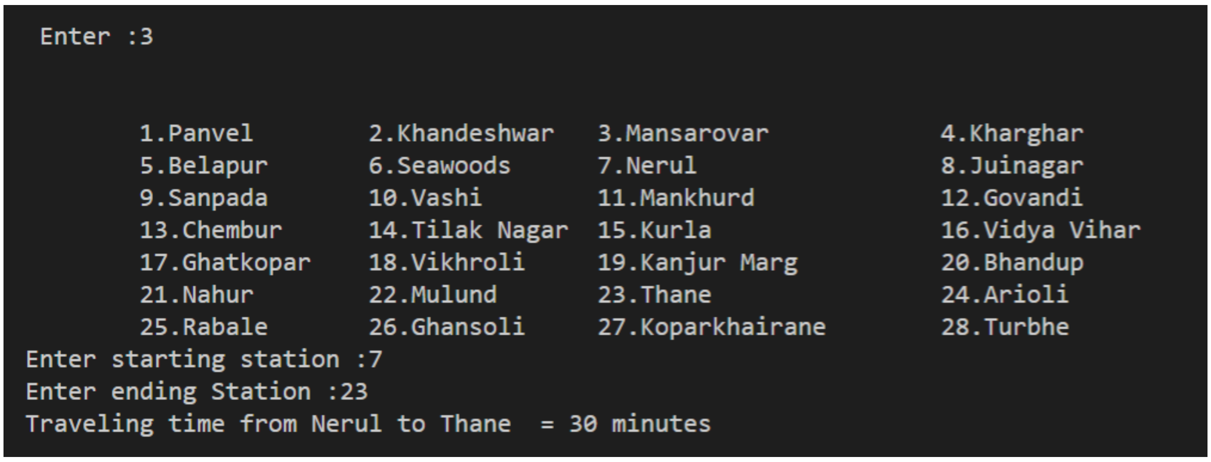
```
 Enter :1


       1.Panvel        2.Khandeshwar   3.Mansarovar            4.Kharghar
       5.Belapur       6.Seawoods      7.Nerul                 8.Juinagar
       9.Sanpada       10.Vashi        11.Mankhurd             12.Govandi
       13.Chembur      14.Tilak Nagar  15.Kurla                16.Vidya Vihar
       17.Ghatkopar    18.Vikhroli     19.Kanjur Marg          20.Bhandup
       21.Nahur        22.Mulund       23.Thane                24.Arioli
       25.Rabale       26.Ghansoli     27.Koparkhairane        28.Turbhe
Enter starting station :10
Enter ending Station :23
Best route for Vashi To Thane :
Vashi->Sanpada->Turbhe->Koparkhairane->Ghansoli->Rabale->Airoli->Thane
```

- Find the distance



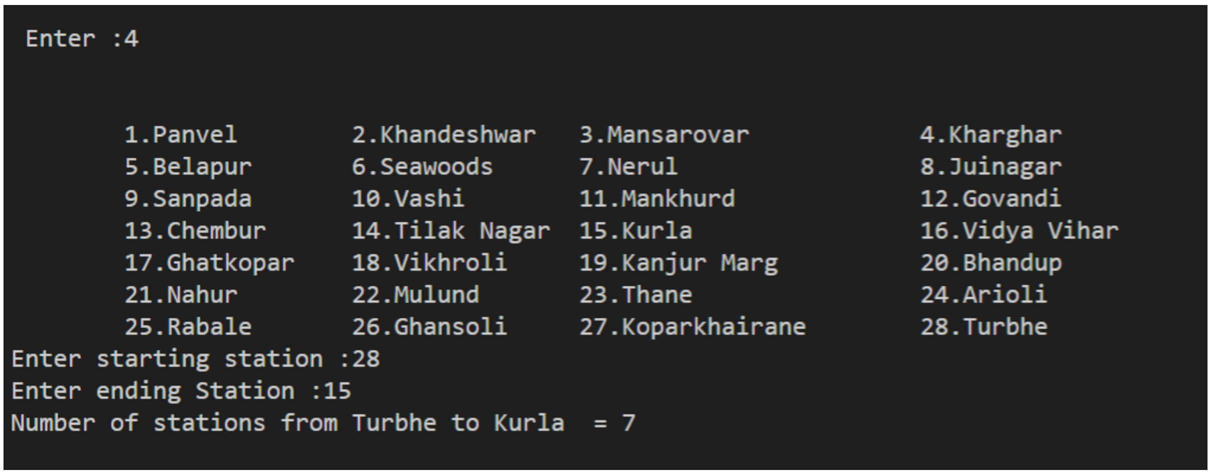
```
 Enter :2


       1.Panvel        2.Khandeshwar   3.Mansarovar            4.Kharghar
       5.Belapur       6.Seawoods      7.Nerul                 8.Juinagar
       9.Sanpada       10.Vashi        11.Mankhurd             12.Govandi
       13.Chembur      14.Tilak Nagar  15.Kurla                16.Vidya Vihar
       17.Ghatkopar    18.Vikhroli     19.Kanjur Marg          20.Bhandup
       21.Nahur        22.Mulund       23.Thane                24.Arioli
       25.Rabale       26.Ghansoli     27.Koparkhairane        28.Turbhe
Enter starting station :1
Enter ending Station :27
Distance from Panvel to Koparkhairane  = 32 km
```

- Find traveling time



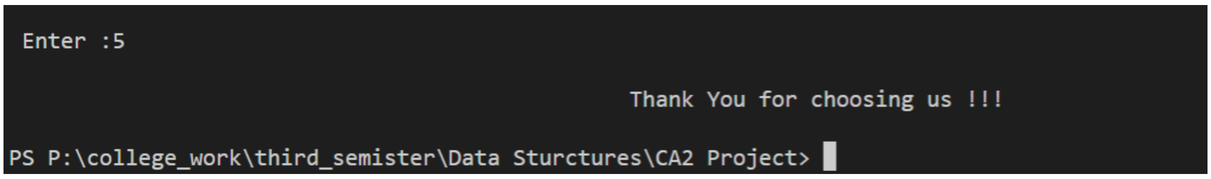
```
 Enter :3


       1.Panvel        2.Khandeshwar   3.Mansarovar            4.Kharghar
       5.Belapur       6.Seawoods      7.Nerul                 8.Juinagar
       9.Sanpada       10.Vashi        11.Mankhurd             12.Govandi
       13.Chembur      14.Tilak Nagar  15.Kurla                16.Vidya Vihar
       17.Ghatkopar    18.Vikhroli     19.Kanjur Marg          20.Bhandup
       21.Nahur        22.Mulund       23.Thane                24.Arioli
       25.Rabale       26.Ghansoli     27.Koparkhairane        28.Turbhe
Enter starting station :7
Enter ending Station :23
Traveling time from Nerul to Thane  = 30 minutes
```

- Count Stations



```
 Enter :4


       1.Panvel        2.Khandeshwar   3.Mansarovar            4.Kharghar
       5.Belapur       6.Seawoods      7.Nerul                 8.Juinagar
       9.Sanpada       10.Vashi        11.Mankhurd             12.Govandi
       13.Chembur      14.Tilak Nagar  15.Kurla                16.Vidya Vihar
       17.Ghatkopar    18.Vikhroli     19.Kanjur Marg          20.Bhandup
       21.Nahur        22.Mulund       23.Thane                24.Arioli
       25.Rabale       26.Ghansoli     27.Koparkhairane        28.Turbhe
Enter starting station :28
Enter ending Station :15
Number of stations from Turbhe to Kurla  = 7
```

- Exit



```
 Enter :5

                                              Thank You for choosing us !!!

PS P:\college_work\third_semister\Data Sturctures\CA2 Project>
```