**Name : Resham Landge**

**Roll No : 2339**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***Assignment No : 6**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Title** : Consider a friends’ network on face book social web site. Model it as a graph to represent each node as a user and a link to represent the fried relationship between them. Store data such as date of birth, number of comments for each user.

1. Find who is having maximum friends

2. Find who has post maximum and minimum comments

3. Find users having birthday in this month.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include<iostream>

#include<stack> //standard library

#include<queue>

using namespace std;

class adjlist

{

public:

typedef struct node //creation of node

{

typedef struct dob //creation of structre for DOB

{

int date,month,year;

}dob;

dob t;

char name[30];

int ncmt,v;

struct node \*next;

}node;

node a[10],\*p[10],\*p1,\*newnode;

int i,j,k,n,m,l,visited[10],cmt[10],w,v1,min,max,m1;

char ans,ans1;

void create();

void display();

void comment();

void bfs(int v1);

void DOB();

};

void adjlist::create() //function to create a network

{

i=0;

do

{

p[i]=new node;

p[i]->v=i;

cout<<"\n\tEnter Name : ";

cin>>p[i]->name;

cout<<"\n\tEnter no of Commentments : ";

cin>>p[i]->ncmt;

cout<<"\n\tEnter Date of Birth : ";

do

{

j=0;

cout<<"\n\tDate : ";

cin>>p[i]->t.date;

if(p[i]->t.date > 31) //if date is greater than 31 then invalid date

{

cout<<"\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Invalid\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\tEnter Again : ";

j=1;

}

}while(j==1);

do

{

j=0;

cout<<"\n\tMonth : ";

cin>>p[i]->t.month;

if(p[i]->t.month > 12) //if month is greater than 12 then invalid month

{

cout<<"\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Invalid\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\tEnter Again : ";

j=1;

}

}while(j==1);

do

{

j=0;

cout<<"\n\tYear : ";

cin>>p[i]->t.year;

if((p[i]->t.year > 2017) || (p[i]->t.year < 1900)) //if year is greater than 2017 & less than 1900 then invalid year

{

cout<<"\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Invalid\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\tEnter Again : ";

j=1;

}

}while(j==1);

cout<<"\n\tDo U want to add more friends (y/n) : ";

cin>>ans;

p[i]->next=NULL;

i++;

}while(ans=='y' || ans=='Y');

n=i;

for(i=0; i<n; i++) //loop to create a friends relation

{

node \*temp=new node;

do

{

cout<<"\n\tEnter Friends of " <<p[i]->name;

cout<<"\n\t1.Yes\n\t2.NO\n\t";

cin>>k;

if(k==1)

{

newnode=new node;

if(p[i]->next==NULL)

p[i]->next=newnode;

else

temp->next=newnode;

temp=newnode;

cout<<"\n\tEnter index of friends : ";

cin>>newnode->v;

}

}while(k==1);

newnode->next=NULL;

}

}

void adjlist::display() //function to display network

{

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

{

cout<<"\n\t"<<p[i]->v;

p1=p[i]->next;

while(p1!=NULL)

{

cout<<"->"<<p1->v;

p1=p1->next;

}

}

}

void adjlist::bfs(int v1) //bfs traversal to find minimum & maximum

{

queue<int> q; //initilize for queue

node \*t1;

for(i=0; i<n; i++) //visited arry is initilize to zero

visited[i]=0;

q.push(v1);

cmt[v1]=p[v1]->ncmt;

visited[v1]=1;

while(!q.empty())

{

v1=q.front(); //dequeue

q.pop();

for(p1=p[v1]; p1=NULL; p1=p1->next)

{

w=p1->v;

if(visited[w]==0) //keep track for mini & Max comment

{

q.push(w);

visited[w]=1;

cmt[v1]=p[v1]->ncmt;

}

}

}

}

void adjlist::comment()

{

min=999;

for(i=0; i<n; i++) //find minimum no of comment

{

if(min > p[i]->ncmt)

{

min=p[i]->ncmt;

j=i;

}

}

cout<<"\n\tMinimum comment in network of "<<p[j]->name<<" has comment "<<p[j]->ncmt;

max=0;

for(i=0; i<n; i++) //find maximum no of comment

{

if(max < p[i]->ncmt)

{

max=p[i]->ncmt;

j=i;

}

}

cout<<"\n\tMaximum comment in network of "<<p[j]->name<<" has comment "<<p[j]->ncmt<<"\n";

}

void adjlist::DOB() //function to find this month date of birth

{

cout<<"\n\tENter the current Month : ";

cin>>m1;

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

{

if(m1==p[i]->t.month)

{

cout<<"\n\tName : "<<p[i]->name;

cout<<"\n\tDOB : "<<p[i]->t.date<<"-"<<p[i]->t.month<<"-"<<p[i]- >t.year<<"\n\n";

}

}

}

int main()

{

adjlist a;

a.create();

a.display();

a.bfs(0);

a.comment();

a.DOB();

}

**Output :**









