***TASK 1:***

#include<iostream>

using namespace std;

class adjacency\_Matrix

{

public:

int\*\* row\_array;

int row;

int col;

adjacency\_Matrix(int Size);

adjacency\_Matrix(int row\_size, int col\_size);

void inserting\_Edge(int num1, int num2);

void display\_Matrix();

};

adjacency\_Matrix::adjacency\_Matrix(int Size)

{

row = Size;

col = Size;

row\_array = new int\* [row];

for (int i = 0; i < row; i++)

{

row\_array[i] = new int[col];

}

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

row\_array[i][j] = 0;

}

}

}

adjacency\_Matrix::adjacency\_Matrix(int row\_size, int col\_size)

{

row = row\_size;

col = col\_size;

row\_array = new int\* [row];

for (int i = 0; i < row; i++)

{

row\_array[i] = new int[col];

}

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

row\_array[i][j] = 0;

}

}

}

void adjacency\_Matrix::inserting\_Edge(int num1, int num2)

{

if (num1 > row || num2 > col || num1 < 0 || num2 < 0)

{

cout << "Invalid Entry" << endl;

}

else

{

row\_array[num1][num2] = 1;

row\_array[num2][num1] = 1;

}

}

void adjacency\_Matrix::display\_Matrix()

{

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

cout << row\_array[i][j] << " ";

}

cout << endl;

}

}

int main()

{

int num1;

int num2;

int rows;

int cols;

cout << "Enter number of rows::";

cin >> rows;

cout << "Enter number of coloumns::";

cin >> cols;

adjacency\_Matrix Obj(rows, cols);

for (int i = 0; i < cols; i++)

{

cout << "Enter Edges::";

cin >> num1 >> num2;

Obj.inserting\_Edge(num1 - 1, num2 - 1);

}

cout << endl;

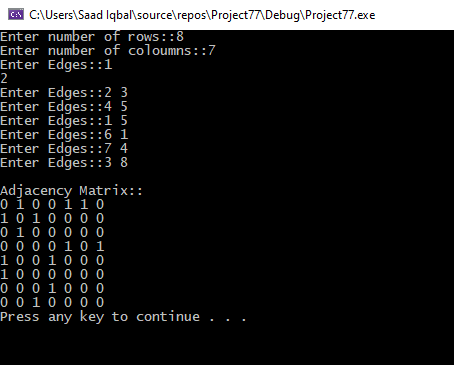
cout << "Adjacency Matrix::" << endl;

Obj.display\_Matrix();

system("pause");

return 0;

}

****

***TASK 2:***

#include<iostream>

using namespace std;

class adjacency\_Matrix

{

public:

int\*\* row\_array;

int row;

int col;

adjacency\_Matrix(int Size);

adjacency\_Matrix(int row\_size, int col\_size);

void inserting\_Edge(int num1, int num2);

void display\_Matrix();

void edge\_count(int num1,int num2);

};

adjacency\_Matrix::adjacency\_Matrix(int Size)

{

row = Size;

col = Size;

row\_array = new int\* [row];

for (int i = 0; i < row; i++)

{

row\_array[i] = new int[col];

}

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

row\_array[i][j] = 0;

}

}

}

adjacency\_Matrix::adjacency\_Matrix(int row\_size, int col\_size)

{

row = row\_size;

col = col\_size;

row\_array = new int\* [row];

for (int i = 0; i < row; i++)

{

row\_array[i] = new int[col];

}

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

row\_array[i][j] = 0;

}

}

}

void adjacency\_Matrix::inserting\_Edge(int num1, int num2)

{

if (num1 > row || num2 > col || num1 < 0 || num2 < 0)

{

cout << "Invalid Entry" << endl;

}

else

{

row\_array[num1][num2] = 1;

row\_array[num2][num1] = 1;

}

}

void adjacency\_Matrix::edge\_count(int num1,int num2)

{

}

void adjacency\_Matrix::display\_Matrix()

{

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

cout << row\_array[i][j] << " ";

}

cout << endl;

}

}

int main()

{

int num1;

int num2;

int rows;

int cols;

int count=0;

cout << "Enter number of rows::";

cin >> rows;

cout << "Enter number of coloumns::";

cin >> cols;

adjacency\_Matrix Obj(rows, cols);

for (int i = 0; i < cols; i++)

{

cout << "Enter Edges::";

cin >> num1 >> num2;

Obj.inserting\_Edge(num1 - 1, num2 - 1);

count++;

}

cout << endl;

cout << "Adjacency Matrix::" << endl;

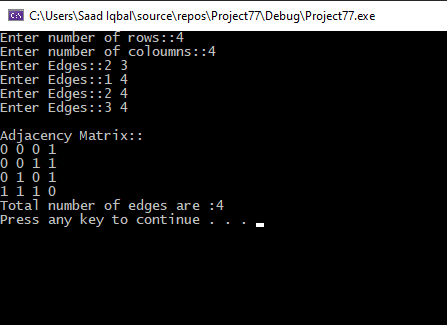
Obj.display\_Matrix();

cout << "Total number of edges are :" << count<<endl;

system("pause");

return 0;

}

******

**TASK 3:**

#include<iostream>

using namespace std;

class adjacency\_Matrix

{

public:

int\*\* row\_array;

int row;

int col;

int \*\*arr;

int result[5][5] = {0};

adjacency\_Matrix(int Size);

adjacency\_Matrix(int row\_size, int col\_size);

void inserting\_Edge(int num1, int num2);

void display\_Matrix();

void input(int row,int col);

void transpose\_mul();

};

adjacency\_Matrix::adjacency\_Matrix(int Size)

{

row = Size;

col = Size;

row\_array = new int\* [row];

for (int i = 0; i < row; i++)

{

row\_array[i] = new int[col];

}

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

row\_array[i][j] = 0;

}

}

}

adjacency\_Matrix::adjacency\_Matrix(int row\_size, int col\_size)

{

row = row\_size;

col = col\_size;

row\_array = new int\* [row];

for (int i = 0; i < row; i++)

{

row\_array[i] = new int[col];

}

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

row\_array[i][j] = 0;

}

}

arr = new int\* [row];

for (int i = 0; i < row; i++)

{

arr[i] = new int[col];

}

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

arr[i][j] = 0;

}

}

}

void adjacency\_Matrix::inserting\_Edge(int num1, int num2)

{

if (num1 > row || num2 > col || num1 < 0 || num2 < 0)

{

cout << "Invalid Entry" << endl;

}

else

{

row\_array[num1][num2] = 1;

row\_array[num2][num1] = 1;

}

}

void adjacency\_Matrix::input(int row, int col)

{

cout << "Enter the values of adjacency matrix " << endl;

for (int j = 0; j < row; j++)

{

for (int i = 0; i < col; i++)

{

cout << "Enter the element [" << j << "][ " << i<<"] :";

cin >> row\_array[j][i];

}

}

}

void adjacency\_Matrix::transpose\_mul()

{

for (int j = 0; j < row; j++)

{

for (int i = 0; i < col; i++)

{

arr[i][j]= row\_array[j][i];

}

}

cout << "Transpose of a matrix" << endl;

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

cout<<arr[i][j]<<" ";

}

cout << endl;

}

for (int i = 0; i < row; i++)

{

cout << i + 1 << " -----> ";

for (int j = 0; j < col; j++)

{

if (row\_array[i][j] == 1)

{

cout << j + 1 << " -----> ";

}

}

cout << endl;

}

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

result[i][j] = 0;

for (int k = 0; k < col; k++)

{

result[i][j] += arr[k][i] \* row\_array[j][k];

}

}

}

cout << endl;

cout << "Result after multiplication :"<<endl;

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

cout << result[i][j]<<" ";

}

cout << endl;

}

}

void adjacency\_Matrix::display\_Matrix()

{

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

cout << row\_array[i][j] << " ";

}

cout << endl;

}

}

int main()

{

int num1;

int num2;

int rows;

int cols;

int count=0;

cout << "Enter number of rows::";

cin >> rows;

cout << "Enter number of coloumns::";

cin >> cols;

adjacency\_Matrix Obj(rows, cols);

Obj.input(rows, cols);

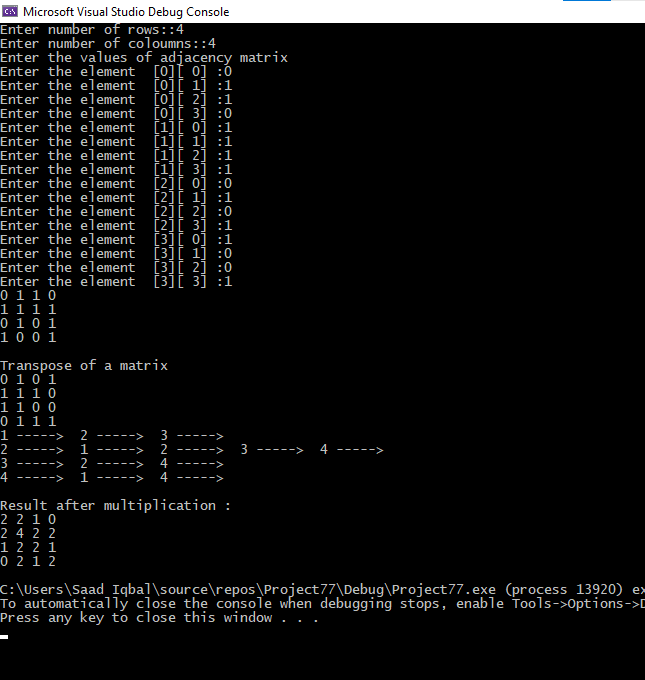
Obj.display\_Matrix();

cout << endl;

Obj.transpose\_mul();

}

**As after multiplication adjacency list of matrix cannot be found. So I’ve perform adjacency operation on the original matrix**

******

***TASK 4:***

#include<iostream>

using namespace std;

struct Node

{

int data;

Node\* next;

};

class Adjacency\_List

{

public:

Node\* head;

};

class Adjacency\_Matrix

{

public:

int\*\* row\_array;

int data;

Adjacency\_List\* obj\_array;

Adjacency\_Matrix(int data);

void print\_adj\_List(int);

void print\_Matrix(int);

void print\_List(int);

void data\_Setting(int num1, int num2, int wieght);

};

Adjacency\_Matrix::Adjacency\_Matrix(int data)

{

data = data;

obj\_array = new Adjacency\_List[data];

for (int i = 0; i < data; ++i)

{

obj\_array[i].head = NULL;

}

row\_array = new int\* [data];

for (int i = 0; i < data; i++)

{

row\_array[i] = new int[data - 1];

for (int j = 0; j < data; j++)

{

row\_array[i][j] = 0;

}

}

}

void Adjacency\_Matrix::print\_List(int a)

{

for (int i = 0; i < a; i++)

{

cout << i + 1;

for (int j = 0; j < a; j++)

{

if (row\_array[i][j] != 0)

{

cout << " (" << row\_array[i][j] << ") --> " << j + 1;

}

}

cout << endl;

}

}

void Adjacency\_Matrix::print\_Matrix(int a)

{

for (int i = 0; i < a; i++)

{

for (int j = 0; j < a; j++)

{

cout << row\_array[i][j] << " ";

}

cout << endl;

}

}

void Adjacency\_Matrix::data\_Setting(int num1, int num2, int weight)

{

row\_array[num1][num2] = weight;

row\_array[num2][num1] = weight;

}

int main()

{

int n = 0, m = 0, a = 0, b = 0;

int option = 0;

int count = 0;

cout << "Enter number of vertices::";

cin >> n;

if (option != 5)

{

cout << "Enter the number of egdes:";

cin >> m;

Adjacency\_Matrix obj(n);

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

{

cout << "Enter the 1st coordinates of " << i + 1 << " egde:";

cin >> a;

while (a > n)

{

cout << "\n\t\tWrong vertice \nEnter again :: ";

cin >> a;

}

cout << "Enter the 2nd coordinates of " << i + 1 << " egde:";

cin >> b;

while (b > n)

{

cout << "\n\t\tWrong vertice \nEnter again :: ";

cin >> b;

}

a -= 1; b -= 1;

cout << "Enter Weight:";

int wei[5];

cin >> wei[i];

obj.data\_Setting(a, b, wei[i]);

for (int j = 0; j < m; j++)

{

if (wei[i] == wei[j+1])

{

count++;

}

}

}

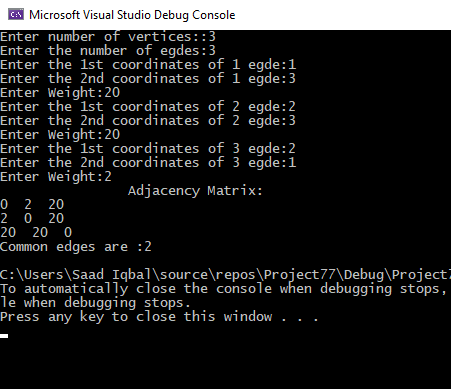
cout << "\t\tAdjacency Matrix:" << endl;

obj.print\_Matrix(n);

cout << "Common edges are :" << count<<endl;

}

}

******