Incomplete Model Checker (A try) :D

MANUAL:

Int=On output screen number of states will be asked :Number of nodes of graph

Int=Then number of prepositions on each node will be asked

Must be >0 and <5

String=prepositions to enter

Int= link between two states (from) s1 (t0) s2… >0 and < states

String =CTL statement to check

Coded in Visual Studio 2017

CODE:

Graph.h

```
#pragma once
#include<iostream>
#include<string>
using namespace std;
const int size1 = 50;
const int o = 50;
const int p = 50;
class graph
{
	int D[size1][size1];
	string A[size1];
	int s,no;
	string ctl;
public:
	void input();
	bool check();
	bool path(int W[o][p], int i, int j);
};
```

graph.cpp

```
#include "pch.h"
#include "graph.h"



void graph::input()
{
	char choice, choice1,choice2;
	do{
		int  v1=0, v2=0;
		string a, b, c, d;
		cout << "Enter the number of states : ";
		cin >> s;

		for (int i = 0; i < s; i++)
		{
			for (int j = 0; j < s; j++)
			{
				D[i][j] = 0;
			}
		}
		cout << "Enter no. of prepositions : ";
		cin >> no;
		if ((no > 0) && (no < 5))
		{

			for (int i = 0; i < s; i++)
```

```
{
        cout << "Enter the prepositions for state: " << i << endl;
        cout << "1st preposition: ";
        cin >> a;
        if (no > 1)
        {
                cout << "2nd preposition: ";
                cin >> b;
                if (no > 2)
                {
                        cout << "3rd preposition: ";
                        cin >> c;
                        if (no == 4)
                        {
                                cout << "4th preposition: ";
                                cin >> d;
                        }
                }
        }

        A[i] = (a + b + c + d);
}
cout << "do you want to create a link between two states? (y/n) ";
cin >> choice;
if (choice != 'n')
{
        do
        {
                cout << "enter state 1: ";
                cin >> v1;
                cout << "enter state 2: ";
```

```
                                        cin >> v2;
                                        if ((v1 >= 0 && v1 < s) && (v2 >= 0 && v2 < s))
                                        {
                                                cout << endl;
                                                D[v1][v2] = 1;
                                        }
                                        else
                                                cout << "State does not exist!! " << endl;
                                        cout << "do you want to create a link between two states?
(y/n) : ";
                                        cin >> choice;
                                } while (choice != 'n');
                        }
                        do
                        {
                                int k = 0;
                                cout << "Enter the CTL statement !!" << endl;
                                cin >> ctl;
                                 if (check())
                                         cout << "True";
                                 else
                                         cout << "False";
                                cout << "\n check another CTL Statement? (y/n) : ";
                                cin >> choice2;
                        } while (choice2 != 'n');
                }
                else
                        cout << "Cannot enter more than 4 prepositions!!" << endl;
                cout << "\n Do you want to check a model? (y/n) : ";
                cin >> choice1;
        } while (choice1 != 'n');
```

```
}



bool graph::check()
{
	int i = 0;
	int W[size1][size1];
	char c;
	for (int i = 0; i < s; i++)
	{
		for (int j = 0; j < s; j++)
		{
			W[i][j] = D[i][j];
		}
	}
	int flag = 0;
	bool change = false, found = false, d = false,Eis=false;
	c = ctl.at(i);
	if ((c == 'E') )
		Eis =true;
	int g = 0;
	for (int b = 0; b < s; b++)
	{
		string u= "  ";
		if (W[g][b] == 1)
		{
			if (ctl.at(i + 1) == 'G')
			{
				for (int f = 0; f < s; f++)
				{
```

```
for (int e = 0; e < s; e++)
{
	if (W[f][e] == 1)
	{
		i = 0;
		c = ctl.at(i + 3);
		if (c == '~')
		{
			if (ctl.at(i + 4) == '(')
			{
				c = ctl.at(i + 5);
				d = true;
				i++;
			}
			else
			{
				u = ctl.at(i + 3);
				u+=ctl.at(i + 4);
			}
			i++;
		}
		if (ctl.at(i + 4) == '-')
		{
			for (int j = 0; j < s; j++)
			{
				for (int t = 0; t < no; t++)
				{
					if ((c == A[j].at(t)) ||
(u.at(0)== A[j].at(t)&& u.at(1)== A[j].at(t+1)))
					{
						if (ctl.at(i +
6) == 'A')
```

```
                                                                                {
                                                                                        if
(ctl.at(i + 7) == 'X')
                                                                                        {

        int m = j;

        for (int n = 0; n < s; n++)

        {

                if (W[m][n] == 1)

                {

                        for (int k = 0; ((k <( A[j].size() ))&& (flag != 1)); k++)

                        {

                                string a="  ";

                                if ((ctl.at(i + 8)) == '~')

                                {


                                        a = ctl.at(i + 8);

                                        a+=ctl.at(i + 9);

                                }

                                else

                                 a = ctl.at(i + 8);
```

```
                                        if ((A[j].at(k) == '~') && (a.at(0) != '~'))

                                        {

                                                k++;

                                        }

                                else if ((a.at(0) == A[n].at(k))||((a.at(0) == A[n].at(k))&&((a.at(1) ==
A[n].at(k+1)))))

                                        {

                                                flag = 1;

                                        }

                                }

                                if ((flag == 0) && (d = false))

                                        return false;

                        }

                }

                found = true;

                                                                                        }

                                                                                        else
if (ctl.at(i + 7) == 'F')

                                                                                        {

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

                {
```

```
for (int n = 0; n < s; n++)
{
        for (int x = 0; x < s; x++)
        {
                if (W[m][n] == 1)
                {
                        if (W[n][x] == 1)
                        {
                                for (int k = 0; ((k < A[j].size()) && (flag != 1)); k++)
                                {
                                        string a="  ";
                                        if (ctl.at(i + 8) == '~')
                                        {
                                                a = ctl.at(i + 8);
                                                a+=ctl.at(i + 9);
                                        }
                                        else
                                                a = ctl.at(i + 8);
                                        if ((A[j].at(k) == '~') && (a.at(0) != '~'))
```

```
                                                {
                                                        k++;
                                                }
                                                else if ((a.at(0) == A[n].at(k)) || ((a.at(0) ==
A[n].at(k)) && ((a.at(1) == A[n].at(k + 1))))
                                                {
                                                        flag = 1;
                                                }
                                        }

                                }

                        }

                }

                if ((flag == 0) && (d == false))
                        return false;
        }
    }
    found = true;
                                                                        }
                                                                }
```

```
                                                                                else if
(ctl.at(i + 6) == 'E')
                                                                                {
                                                                                        if
(ctl.at(i + 7) == 'X')
                                                                                        {


        int m = j;
        for (int n = 0; (n < s) && (found != true); n++)
        {
                if (W[m][n] == 1)
                {
                        for (int k = 0; ((k < A[j].size()) && (flag != 1)); k++)
                        {
                                string a="  ";
                                if (ctl.at(i + 8) == '~')
                                {
                                        a = ctl.at(i + 8);
                                        a+=ctl.at(i + 9);
                                }
                                else
                                        a = ctl.at(i + 8);
```

```
                                        if ((A[j].at(k) == '~') && (a.at(0) != '~'))
                                        {
                                                k++;
                                        }
                                    else if ((a.at(0) == A[n].at(k)) || ((a.at(0) == A[n].at(k)) && ((a.at(1) ==
A[n].at(k + 1)))))
                                        {
                                                flag = 1;
                                                found = true;
                                        }
                                }
                        }
                }
        if ((d == false) && (flag == 0))
                return false;
                                                                                        }
                                                                                        else
if (ctl.at(i + 7) == 'F')
                                                                                        {
        found = false;
        for (int m = 0; m < s && found != true; m++)
```

```
{
        for (int n = 0; n < s && found != true; n++)
        {
                for (int x = 0; x < s; x++)
                {
                        if (W[m][n] == 1)
                        {
                                if (W[n][x] == 1)
                                {
                                        for (int k = 0; ((k <A[j].size()) && (flag != 1)); k++)
                                        {
                                                string a="  ";
                                                if (ctl.at(i + 8) == '~')
                                                {
                                                        a = ctl.at(i + 8);
                                                        a+=ctl.at(i + 9);
                                                }
                                                else
                                                        a = ctl.at(i + 8);
```

```
                                        if ((A[j].at(k) == '~') && (a.at(0) != '~'))
                                        {
                                                k++;
                                        }
                                        else if ((a.at(0) == A[n].at(k)) || ((a.at(0) ==
A[n].at(k)) && ((a.at(1) == A[n].at(k + 1)))))
                                        {
                                                flag = 1;
                                                found = true;
                                        }
                                }
                        }
                }
                        }
                }
        }
        if ((d == false) && (flag == 0))
                return false;
                                                                        }
```

```
                                                                                }
                                                                                else
                                                                                {
                                                                                        for
(int k = 0; (k < A[j].size() && (flag != 1)); k++)
                                                                                        {

        string a="  ";
        if (ctl.at(i + 6) == '~')
        {
                a = ctl.at(i + 6);
                a+=ctl.at(i + 7);
        }
        else
                 a = ctl.at(i + 6);
        if ((A[j].at(k) == '~') && (a.at(0) != '~'))
        {
                k++;
        }
        else if((a.at(0) == A[j].at(k)) || ((a.at(0) == A[j].at(k)) && ((a.at(1) == A[j].at(k + 1)))))
        {
                flag = 1;
```

```
                found = true;

        }
                                                                                }
                                                                                if
((d == false) && (flag == 0))

        return false;
                                                                        }
                                                                }
                                                        }
                                                }

                                                        }
                                                        i = 0;
                                                }
                                                        if ((found == true) && (d == true) && (ctl.at(0) !=
'E'))
                                                                return false;
                                                }
                                        }
                                }
                                else if (ctl.at(i + 1) == 'F')
                                {
                                        for (int f = 0; f < s; f++)
                                        {
                                                for (int e = 0; e < s; e++)
                                                {
                                                        if (W[f][e] == 1)
                                                        {
                                                                i = 0;
                                                                c = ctl.at(i + 3);
```

```
if (c == '~')
{
	if (ctl.at(i + 4) == '(')
	{
		c = ctl.at(i + 5);
		d = true;
		i++;
	}
	else
	{
		u = ctl.at(i + 3);
		u+=ctl.at(i + 4);
	}
	i++;
}
if (ctl.at(i + 4) == '-')
{
	for (int j = 0; j < s; j++)
	{
		for (int t = 0; t < no; t++)
		{
			if ((c == A[j].at(t)) ||
(u.at(0) == A[j].at(t) && u.at(1) == A[j].at(t + 1)))

6) == 'A')

(ctl.at(i + 7) == 'X')

	int m = 0;
```

```
for (int n = 0; n < s; n++)
{
        if (W[m][n] == 1)
        {
                for (int k = 0; ((k < A[j].size()) && (flag != 1)); k++)
                {
                        string a="  ";
                        if ((ctl.at(i + 8)) == '~')
                        {
                                a = ctl.at(i + 8);
                                a+=ctl.at(i + 9);
                        }
                        else
                                a = ctl.at(i + 8);
                        if ((A[j].at(k) == '~') && (a.at(0) != '~'))
                        {
                                k++;
                        }
```

```
                                else if ((a.at(0) == A[n].at(k)) || ((a.at(0) == A[n].at(k)) && ((a.at(1)
== A[n].at(k + 1)))))

                                {

                                        found = true;

                                        flag = 1;

                                }

                        }

                        if ((flag == 0) && (d == false))

                                found = false;

                }

        }

        if ((d == true) && (found = true))

                found = false;
                                                                                        }
                                                                                        else
if (ctl.at(i + 7) == 'F')
                                                                                        {

        flag = 0;

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

        {

                for (int n = 0; n < s; n++)
```

```
{
        if (W[m][n] == 1)
        {
            for (int k = 0; ((k < A[j].size()) && (flag != 1)); k++)
            {
                string a="  ";
                if ((ctl.at(i + 8)) == '~')
                {
                    a = ctl.at(i + 8);
                    a+=ctl.at(i + 9);
                }
                else
                    a = ctl.at(i + 8);
                if ((A[j].at(k) == '~') && (a.at(0) != '~'))
                {
                    k++;
                }
                else if ((a.at(0) == A[n].at(k)) || ((a.at(0) == A[n].at(k)) &&
((a.at(1) == A[n].at(k + 1)))))
```

```
                                        {
                                                found = true;
                                                flag = 1;
                                        }
                                }
                        }
                }
        }
        if ((flag == 0))
                found = false;
        if ((found = true) && (d == true))
                found = false;
                                                                        }
                                                                }
                                                                else if
(ctl.at(i + 6) == 'E')
                                                                {
                                                                        if
(ctl.at(i + 7) == 'X')
                                                                        {


        int m = j;
        for (int n = 0; n < s && found != true; n++)
```

```
{
        if (W[m][n] == 1)
        {
            for (int k = 0; ((k < A[j].size()) && (flag != 1)); k++)
            {
                string a;
                if ((ctl.at(i + 8)) == '~')
                {
                    a = ctl.at(i + 8);
                    a+=ctl.at(i + 9);
                }
                else
                     a = ctl.at(i + 8);
                if ((A[j].at(k) == '~') && (a.at(0) != '~'))
                {
                    k++;
                }
                else if ((a.at(0) == A[n].at(k)) || ((a.at(0) == A[n].at(k)) && ((a.at(1)
== A[n].at(k + 1))))
```

```
                                        {
                                                found = true;
                                                flag = 1;
                                        }
                                }
                        }
                if ((d == true) && (found == true))
                        found = false;
                                                                                }
                                                                                else
if (c == 'F')
                                                                                {
                for (int m = 0; (m < s) && (found != true); m++)
                {
                        for (int n = 0; (n < s) && (found != true); n++)
                        {
                                if (W[m][n] == 1)
                                {
                                        for (int k = 0; ((k < A[j].size()) && (flag != 1)); k++)
```

```
{
        string a;
        if ((ctl.at(i + 8)) == '~')
        {
                a = ctl.at(i + 8);
                a+=ctl.at(i + 9);
        }
        else
                a = ctl.at(i + 8);
        if ((A[j].at(k) == '~') && (a.at(0) != '~'))
        {
                k++;
        }
        else if ((a.at(0) == A[n].at(k)) || ((a.at(0) == A[n].at(k)) &&
((a.at(1) == A[n].at(k + 1)))))
        {
                found = true;
                flag = 1;
        }
```

```
                                        }

                                }

                        }

                }

        if ((d == true) && (found = true))

                found = false;

                                                                        }
                                                                }
                                                                else
                                                                {
                                                                        for
(int k = 0; ((k < A[j].size()) && (flag != 1)); k++)
                                                                        {

        string a="  ";

        if ((ctl.at(i + 6)) == '~')

        {

                a = ctl.at(i + 6);

                a+=ctl.at(i + 7);

        }

        else

                 a = ctl.at(i + 6);
```

```
        if ((A[j].at(k) == '~') && (a.at(0) != '~'))
        {
                k++;
        }
        else if ((a.at(0) == A[j].at(k)) || ((a.at(0) == A[j].at(k)) && ((a.at(1) == A[j].at(k + 1)))))
        {
                found = true;
                flag = 1;
        }
                                                                                }
                                                                        }
                                                                        if ((d ==
true) && (found = true))
        found = false;
                                                                }
                                                        }
                                                }

                                        }
                                        if (found == false && (ctl.at(i) != 'E'))
                                                return false;
                                }
                        }
                }
        }
```

```
else if (ctl.at(i + 1) == 'X')
{
	int f = 0;
	for (int e = 0; e < s; e++)
	{
		if (W[f][e] == 1)
		{
			i = 0;
			c = ctl.at(i + 3);
			if (c == '~')
			{
				if (ctl.at(i + 4) == '(')
				{
					c = ctl.at(i + 5);
					d = true;
					i++;
				}
				else
				{
					u = ctl.at(i + 3);
					u+=ctl.at(i + 4);
				}
				i++;
			}
			if (ctl.at(i + 4) == '-')
			{
				for (int j = 0; j < s; j++)
				{
					for (int t = 0; t < no; t++)
					{
```

```
                                                                    if ((c == A[j].at(t)) || (u.at(0)
== A[j].at(t) && u.at(1) == A[j].at(t + 1)))
                                                                    {
                                                                        if (ctl.at(i + 6) == 'A')
                                                                        {
                                                                            if (ctl.at(i +
7) == 'X')
                                                                            {
                                                                                int
m = j;
                                                                                for
(int n = 0; n < s; n++)
                                                                                {
        if (W[m][n] == 1)
        {
            for (int k = 0; ((k < A[j].size()) && (flag != 1)); k++)
            {
                string a="  ";
                if ((ctl.at(i + 8)) == '~')
                {
                    a = ctl.at(i + 8);
                    a+=ctl.at(i + 9);
                }
                else
                    a = ctl.at(i + 8);
```

```
                        if ((A[j].at(k) == '~') && (a.at(0) != '~'))
                        {
                                k++;
                        }
                        else if ((a.at(0) == A[n].at(k)) || ((a.at(0) == A[n].at(k)) && ((a.at(1) ==
A[n].at(k + 1)))))
                        {
                                found = true;
                                flag = 1;
                        }
                }
                if ((flag == 0) && (d == false))
                        found = false;
        }
                                                                                        }
                                                                                        if
((d == true) && (found = true))
        found = false;
                                                                                }
                                                                                else if
(ctl.at(i + 7) == 'F')
```

```
                                                                                {
                                                                                    for
(int m = 0; m < s; m++)
                                                                                    {

        for (int n = 0; n < s; n++)

        {

                if (W[m][n] == 1)

                {

                        for (int k = 0; ((k < A[j].size()) && (flag != 1)); k++)

                        {

                                string a="  ";

                                if ((ctl.at(i + 8)) == '~')

                                {

                                        a = ctl.at(i + 8);

                                        a+=ctl.at(i + 9);

                                }

                                else

                                         a = ctl.at(i + 8);

                                if ((A[j].at(k) == '~') && (a.at(0) != '~'))

                                {
```

```
                    k++;
                }
                else if ((a.at(0) == A[n].at(k)) || ((a.at(0) == A[n].at(k)) && ((a.at(1)
== A[n].at(k + 1))))
                {
                    found = true;
                    flag = 1;
                }
            }

        }
    }
                                                                    }
                                                                    if
((flag == 1))
    found = true;
                                                                    else
    found = false;
                                                                    if
((found = true) && (d == true))
    found = false;
                                                                }
                                                            }
```

```
                                                                    else if (ctl.at(i + 6)
== 'E')
                                                                    {
                                                                        if (ctl.at(i +
7) == 'X')
                                                                        {


                                                                            int
m = 0;
                                                                            for
(int n =0; n < s && found != true; n++)
                                                                            {

        if (W[m][n] == 1)

        {

            for (int k = 0; ((k < A[j].size()) && (flag != 1)); k++)

            {

                string a="  ";

                if ((ctl.at(i + 8)) == '~')

                {

                    a = ctl.at(i + 8);

                    a+=ctl.at(i + 9);

                }

                else

                    a = ctl.at(i + 8);
```

```
                        if ((A[j].at(k) == '~') && (a.at(0) != '~'))
                        {
                                k++;
                        }
                        else if ((a.at(0) == A[n].at(k)) || ((a.at(0) == A[n].at(k)) && ((a.at(1) ==
A[n].at(k + 1)))))
                        {
                                found = true;
                                flag = 1;
                        }
                }

        }
                                                                        }
                                                                        if
((d == true) && (found == true))
        return false;
                                                                }
                                                                else if
(ctl.at(i + 7) == 'F')
                                                                {
                                                                        for
(int m = 0; (m < s) && (found != true); m++)
                                                                        {
```

```
for (int n = 0; (n < s) && (found != true); n++)
{
        if (W[m][n] == 1)
        {
                for (int k = 0; ((k < A[j].size()) && (flag != 1)); k++)
                {
                        string a="  ";
                        if ((ctl.at(i + 8)) == '~')
                        {
                                a = ctl.at(i + 8);
                                a+=ctl.at(i + 9);
                        }
                        else
                                a = ctl.at(i + 8);
                        if ((A[j].at(k) == '~') && (a.at(0) != '~'))
                        {
                                k++;
                        }
```

```
                                else if ((a.at(0) == A[n].at(k)) || ((a.at(0) == A[n].at(k)) && ((a.at(1)
== A[n].at(k + 1)))))

                                {

                                        found = true;

                                        flag = 1;

                                }

                        }


                }

        }
                                                                                        }
                                                                                        if
((d == true) && (found = true))

        found = false;
                                                                                }
                                                                        }
                                                                        else
                                                                        {
                                                                                for (int k =
0; ((k < A[j].size()) && (flag != 1)); k++)
                                                                                {

        string a="  ";
                                                                                        if
((ctl.at(i + 6)) == '~')
                                                                                        {
```

```
            a = ctl.at(i + 6);

            a+=ctl.at(i + 7);
                                                                                    }
                                                                                    else

             a = ctl.at(i + 6);
                                                                                    if
((A[j].at(k) == '~') && (a.at(0) != '~'))
                                                                                    {

            k++;
                                                                                    }
                                                                                    else
if ((a.at(0) == A[j].at(k)) || ((a.at(0) == A[j].at(k)) && ((a.at(1) == A[j].at(k + 1)))))
                                                                                    {

            found = true;

            flag = 1;
                                                                                    }
                                                                                }
                                                                                if ((d ==
true) && (found = true))

            found = false;;
                                                                                        }
                                                                                    }
                                                                                }
                                                                            }
                                                                        }
                                                                        if ((found == false) && (Eis==false))
                                                                            return false;
```

```
                                        }


                                }

                        }

                        if ((Eis==true) && (found == false))

                                return false;

                }

        }

        return true;

}
```

Source.cpp

```
#include"pch.h"
#include"graph.h"
using namespace std;
int main()
{
        graph g;
        g.input();
}
```