ARTIFICIAL INTELLIGENCE
(AI)

BACHELOR's OF TECHNOLOGY

(Computer Science Engineering)

SUB MITTED BY: -

Pariva

Roll no: -1803546

Semester: - 7th

CTIEMT

SUBMITTED TO: -

Mr Abhishek Bhardwaj

(**Assistant professor)**

CSE/IT dept.

Table of Contents

Practical: 1

AIM: - Write a program for Tower of Hanoi.

```
#include<iostream>
#include<conio.h>
void towerHanoi(int n,char mid,char end,char beg)
{
   if(n==1){
      std::cout<<" move disc from"<<beg<<"------"<<end<<'\n';
   }
   else
   {
      towerHanoi(n-1,beg,end,mid);
      std::cout<<"move disc from"<<beg<<"-------"<<end<<'\n';
      towerHanoi(n-1,mid,beg,end);
   }
}
int main()
{

   int n;
   std::cout<<"enter the number of disc you want to display:\n";
   std::cin>>n;
   towerHanoi(n,'A','B','C');
   getch();
}
```

Output: - When n=5

```
enter the number of disc you want to display:
5
 move disc fromC------B
move disc fromA-------B
 move disc fromB------A
move disc fromC-------B
 move disc fromA------C
move disc fromB-------C
 move disc fromC------B
move disc fromA-------B
 move disc fromB------A
move disc fromC-------A
 move disc fromA------C
move disc fromB-------A
 move disc fromC------B
move disc fromA-------B
 move disc fromB------A
move disc fromC-------B
 move disc fromA------C
move disc fromB-------C
 move disc fromC------B
move disc fromA-------C
 move disc fromB------A
move disc fromC-------A
 move disc fromA------C
move disc fromB-------C
 move disc fromC------B
move disc fromA-------B
 move disc fromB------A
move disc fromC-------B
```

```
 move disc fromA-------C
move disc fromB--------C
 move disc fromC-------B
move disc fromA--------C
 move disc fromB------A
move disc fromC--------A
 move disc fromA-------C
move disc fromB--------C
 move disc fromC-------B
move disc fromA--------B
 move disc fromB-------A
move disc fromC--------B
 move disc fromA-------C
move disc fromB--------C
 move disc fromC-------B


...Program finished with exit code 0
Press ENTER to exit console.
```

When n=4

```
enter the number of disc you want to display:
4
 move disc fromA-------B
move disc fromC--------B
 move disc fromB-------C
move disc fromA--------B
 move disc fromC------A
move disc fromB--------A
 move disc fromA-------B
move disc fromC--------B
 move disc fromB-------C
move disc fromA--------C
 move disc fromC------A
move disc fromB--------C
 move disc fromA------B
move disc fromC--------B
 move disc fromB------C


...Program finished with exit code 0
Press ENTER to exit console.
```

When n=3

```
enter the number of disc you want to display:
3
 move disc fromC------B
move disc fromA-------B
 move disc fromB------A
move disc fromC-------B
 move disc fromA------C
move disc fromB-------C
 move disc fromC------B


...Program finished with exit code 0
Press ENTER to exit console.
```

When n=2

```
enter the number of disc you want to displa
2
 move disc fromA------B
move disc fromC-------B
 move disc fromB------C


...Program finished with exit code 0
Press ENTER to exit console.
```

Practical: 2

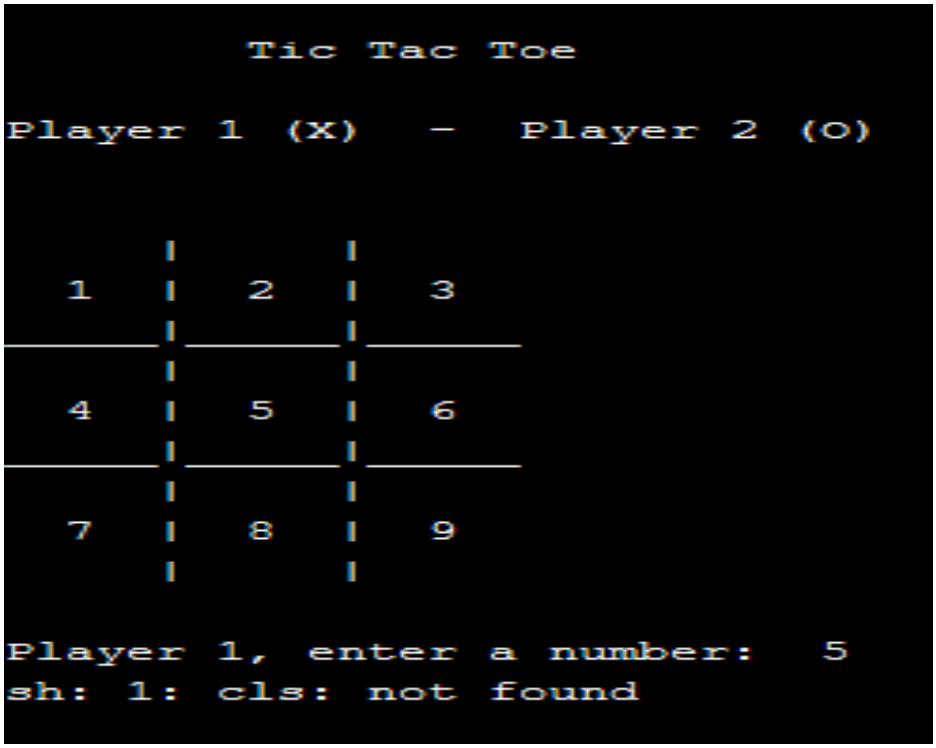
AIM: - Write a program of TIC TAC TOC in C++.

```
#include <iostream>
using namespace std;
char square[10] = {'o','1','2','3','4','5','6','7','8','9'};
int checkwin();
void board();
int main()
{
        int player = 1,i,choice;

   char mark;
   do
   {
      board();
      player=(player%2)?1:2;
      cout << "Player " << player << ", enter a number:  ";
      cin >> choice;
      mark=(player == 1) ? 'X' : 'O';
      if (choice == 1 && square[1] == '1')
         square[1] = mark;
      else if (choice == 2 && square[2] == '2')
         square[2] = mark;
      else if (choice == 3 && square[3] == '3')
         square[3] = mark;
      else if (choice == 4 && square[4] == '4')
 square[4] = mark;
      else if (choice == 5 && square[5] == '5')
         square[5] = mark;
      else if (choice == 6 && square[6] == '6')
         square[6] = mark;
      else if (choice == 7 && square[7] == '7')
         square[7] = mark;
      else if (choice == 8 && square[8] == '8')
         square[8] = mark;
      else if (choice == 9 && square[9] == '9')
         square[9] = mark;
      else
      {
         cout<<"Invalid move ";

         player--;
         cin.ignore();
         cin.get();
      }
      i=checkwin();

      player++;
```

```
    }while(i==-1);
    board();
    if(i==1)
        cout<<"==>\aPlayer "<<--player<<" win ";
    else
        cout<<"==>\aGame draw";
    cin.ignore();
    cin.get();
    return 0;
}

int checkwin()
{
    if (square[1] == square[2] && square[2] == square[3])

        return 1;
    else if (square[4] == square[5] && square[5] == square[6])

        return 1;
    else if (square[7] == square[8] && square[8] == square[9])

        return 1;
    else if (square[1] == square[4] && square[4] == square[7])

        return 1;
    else if (square[2] == square[5] && square[5] == square[8])

        return 1;
    else if (square[3] == square[6] && square[6] == square[9])

        return 1;
    else if (square[1] == square[5] && square[5] == square[9])

        return 1;
    else if (square[3] == square[5] && square[5] == square[7])

        return 1;
    else if (square[1] != '1' && square[2] != '2' && square[3] != '3'
                && square[4] != '4' && square[5] != '5' && square[6] != '6'
               && square[7] != '7' && square[8] != '8' && square[9] != '9')

        return 0;
    else
        return -1;
}
void board()
{
    system("cls");
    cout << "\n\n\tTic Tac Toe\n\n";
    cout << "Player 1 (X)  -  Player 2 (O)" << endl << endl;
```

```
    cout << endl;
    cout << "     |     |     " << endl;
    cout << "  " << square[1] << "  |  " << square[2] << "  |  " << square[3] << endl;
    cout << "_____|_____|_____" << endl;
    cout << "     |     |     " << endl;
    cout << "  " << square[4] << "  |  " << square[5] << "  |  " << square[6] << endl;
    cout << "_____|_____|_____" << endl;
    cout << "     |     |     " << endl;
    cout << "  " << square[7] << "  |  " << square[8] << "  |  " << square[9] << endl;
    cout << "     |     |     " << endl << endl;
}
```

Output:-



```
          Tic Tac Toe

Player 1 (X)  -  Player 2 (O)


     |     |
  1  |  2  |  3
_____|_____|_____
     |     |
  4  |  X  |  6
_____|_____|_____
     |     |
  7  |  8  |  9
     |     |
```

```
Player 2, enter a number:  3
sh: 1: cls: not found


          Tic Tac Toe

Player 1 (X)  -  Player 2 (O)


     |     |
  1  |  2  |  O
_____|_____|_____
     |     |
  4  |  X  |  6
_____|_____|_____
     |     |
  7  |  8  |  9
     |     |
```

```
Player 1, enter a number:  1
sh: 1: cls: not found


          Tic Tac Toe

Player 1 (X)  -  Player 2 (O)


     |     |
  X  |  2  |  O
_____|_____|_____
     |     |
  4  |  X  |  6
_____|_____|_____
     |     |
  7  |  8  |  9
     |     |
```

```
Player 2, enter a number:  2
sh: 1: cls: not found


          Tic Tac Toe

Player 1 (X)  -  Player 2 (O)


     |     |
  X  |  O  |  O
_____|_____|_____
     |     |
  4  |  X  |  6
_____|_____|_____
     |     |
  7  |  8  |  9
     |     |
```

```
Player 1, enter a number:  4
sh: 1: cls: not found


          Tic Tac Toe

Player 1 (X)  -  Player 2 (O)


     |     |
  X  |  O  |  O
_____|_____|_____
     |     |
  X  |  X  |  6
_____|_____|_____
     |     |
  7  |  8  |  9
     |     |
```

```
Player 2, enter a number:  6
sh: 1: cls: not found


          Tic Tac Toe

Player 1 (X)  -  Player 2 (O)


     |     |
  X  |  O  |  O
_____|_____|_____
     |     |
  X  |  X  |  O
_____|_____|_____
     |     |
  7  |  8  |  9
     |     |
```

```
Player 1, enter a number:  7
sh: 1: cls: not found


         Tic Tac Toe

Player 1 (X)  -  Player 2 (O)


     |     |
  X  |  O  |  O
_____|_____|_____
     |     |
  X  |  X  |  O
_____|_____|_____
     |     |
  X  |  8  |  9
     |     |

==>Player 1 win

...Program finished with exit code 9
Press ENTER to exit console.
```

Practical: 3

AIM: - Write a program for Water Jug Problem.

```
#include<bits/stdc++.h>
using namespace std;
int x;
int y;
void show(int a, int b);
int min(int w, int z)
{
if (w < z)
return w;
else
return z;
}
void show(int a, int b)
{
cout << setw(12) << a << setw(12) << b<<endl;
}
void s(int n)
{
int xq = 0, yq = 0;
int t;
cout << setw(15) <<"FIRST JUG"<< setw(15) <<"SECOND JUG"<<endl;
while (xq != n && yq!=n )
{
  if (xq == 0)
  {
    xq = x;
    show(xq, yq);
  }
  else if (yq == y)
  {
    yq = 0;
    show(xq, yq);
  }
  else
  {
    t = min(y - yq, xq);
    yq= yq + t;
    xq = xq - t;
    show(xq, yq);
      }
}
}
int main()
{
int n;
cout << "Enter the liters of water required out of the two jugs: ";
```

```
cin >> n;
cout << "Enter the capacity of the first jug: ";
cin >> x;
cout << "Enter the capacity of the second jug: ";
cin >> y;
if(n<x || n<y)
{ if(n%(__gcd(x,y))==0)
   s(n);
  else
  cout<<"This is not possible....\n";
}
else
  cout<<"This is not possible....\n";
}
```

Output:-

```
Enter the liters of water required out of the two jugs: 2
Enter the capacity of the first jug: 4
Enter the capacity of the second jug: 3
      FIRST JUG     SECOND JUG
           4            0
           1            3
           1            0
           0            1
           4            1
           2            3


...Program finished with exit code 0
Press ENTER to exit console.
```

Practical: 4

AIM: - Write a program to implement Depth First Search (DFS) in C++.

```
#include <iostream>
#include <conio.h>
using namespace std;
int c = 0;
struct node
{
   char data;
   int st_time, lv_time;
}*p = NULL, *r = NULL;
struct stack
{
   node *pt;
   stack *next;
}*top = NULL, *q = NULL, *np= NULL;
void push(node *ptr)
{
   np = new stack;
   np->pt = ptr;
   np->next = NULL;
   if (top == NULL)
   {
      top = np;
   }
   else
   {
      np->next = top;
      top = np;
   }
}
node *pop()
{
   if (top == NULL)
   {
      cout<<"underflow\n";
   }
   else
   {
      q = top;
      top = top->next;
      return(q->pt);
      delete(q);
   }
}
void create(int a[], int b[][7], int i, int j)
{
   c++;
```

```
    p = new node;
    cout<<"enter data for new node\n";
    cin>>p->data;
    p->st_time = c;
    cout<<"start time for "<<p->data<<" is "<<c<<endl;
    a[i] = 1;
    push(p);
    while (j < 7)
    {
        if ((b[i][j] == 0) || (b[i][j] == 1 && a[j] == 1))
        {
            j++;
        }
        else if (b[i][j] == 1 && a[j] == 0)
        {
            create(a,b,j,0);
        }
    }
    r = pop();
    cout<<"node popped\n";
    c++;
    cout<<"leave time for "<<r->data<<" is "<<c<<endl;
    return;
}
int main()
{
    int a[7];
    for (int i = 0; i < 7; i++)
    {
        a[i] = 0;
    }
    int b[7][7];
    cout<<"enter values for adjacency matrix"<<endl;
    for (int i = 0 ; i < 7 ; i++ )
    {
        cout<<"enter values for "<<(i+1)<<" row"<<endl;
        for (int j = 0; j < 7; j++)
        {
            cin>>b[i][j];
        }
    }
    create(a,b,0,0);
    getch();
}
```

Output:-

```
enter values for adjacency matrix
enter values for 1 row
0
1
1
0
0
1
1
enter values for 2 row
1
0
0
0
0
0
0
enter values for 3 row
1
0
0
0
0
0
1
enter values for 4 row
0
0
0
0
1
1
0
```

```
enter values for 5 row
0
0
0
1
0
1
1
enter values for 6 row
1
0
0
1
1
0
1
enter values for 7 row
1
0
1
0
1
1
0
enter data for new node
a
start time for a is 1
enter data for new node
1
start time for 1 is 2
node popped
leave time for 1 is 3
enter data for new node
b
```

```
start time for 2 is 5
enter data for new node
d
start time for d is 6
enter data for new node
e
start time for e is 7
enter data for new node
f
start time for f is 8
node popped
leave time for f is 9
node popped
leave time for e is 10
node popped
leave time for d is 11
node popped
leave time for 2 is 12
node popped
leave time for b is 13
node popped
leave time for a is 14


...Program finished with exit code 0
Press ENTER to exit console.
```

Practical:5

AIM:-WAP to implement Best First Search(BFS) in C++.

```
#include <bits/stdc++.h>
using namespace std;
typedef pair<int, int> pi;
vector<vector<pi> > graph;
void addedge(int x, int y, int cost)
{
   graph[x].push_back(make_pair(cost, y));
   graph[y].push_back(make_pair(cost, x));
}
void best_first_search(int source, int target, int n)
{
   vector<bool> visited(n, false);
   priority_queue<pi, vector<pi>, greater<pi> > pq;
   pq.push(make_pair(0, source));
   visited = true;
   while (!pq.empty()) {
      int x = pq.top().second;
      cout << x << " ";
      pq.pop();
      if (x == target)
         break;
      for (int i = 0; i < graph[x].size(); i++) {
         if (!visited[graph[x][i].second]) {
            visited[graph[x][i].second] = true;
            pq.push(graph[x][i]);
         }
      }
   }
}
int main()
{
   int v = 14;
   graph.resize(v);
   addedge(0, 1, 3);
   addedge(0, 2, 6);
   addedge(0, 3, 5);
   addedge(1, 4, 9);
   addedge(1, 5, 8);
   addedge(2, 6, 12);
   addedge(2, 7, 14);
   addedge(3, 8, 7);
```

```
    addedge(8, 9, 5);
    addedge(8, 10, 6);
    addedge(9, 11, 1);
    addedge(9, 12, 10);
    addedge(9, 13, 2);
    int source = 0;
    int target = 9;
    best_first_search(source, target, v);
    return 0;
}
```

Output-:

```
0 1 3 2 8 9
Process returning θ (θXθ)   Execution time : 32.869 s
Press any key to continue.
```

Practical:6

AIM-: Write a program to implement 3*3 puzzle in C++.

```
#include <bits/stdc++.h>
using namespace std;
#define N 3
struct Node
{
   Node* parent;
   int mat[N][N];
   int x, y;
   int cost;
   int level;
};
int printMatrix(int mat[N][N])
{
   for (int i = 0; i < N; i++)
   {
      for (int j = 0; j < N; j++)
         printf("%d ", mat[i][j]);
      printf("\n");
   }
}
Node* newNode(int mat[N][N], int x, int y, int newX,
            int newY, int level, Node* parent)
{
   Node* node = new Node;
   node->parent = parent;
   memcpy(node->mat, mat, sizeof node->mat);
   swap(node->mat[x][y], node->mat[newX][newY]);
   node->cost = INT_MAX;
   node->level = level;
   node->x = newX;
   node->y = newY;
   return node;
}
int row[] = { 1, 0, -1, 0 };
int col[] = { 0, -1, 0, 1 };
int calculateCost(int initial[N][N], int final[N][N])
{
   int count = 0;
   for (int i = 0; i < N; i++)
     for (int j = 0; j < N; j++)
       if (initial[i][j] && initial[i][j] != final[i][j])
          count++;
   return count;
}
int isSafe(int x, int y)
{
   return (x >= 0 && x < N && y >= 0 && y < N);
```

```
}
void printPath(Node* root)
{
   if (root == NULL)
      return;
   printPath(root->parent);
   printMatrix(root->mat);
   printf("\n");
}
struct comp
{
   bool operator()(const Node* lhs, const Node* rhs) const
   {
      return (lhs->cost + lhs->level) > (rhs->cost + rhs->level);
   }
};
void solve(int initial[N][N], int x, int y,
         int final[N][N])
{
   priority_queue<Node*, std::vector<Node*>, comp> pq;
   Node* root = newNode(initial, x, y, x, y, 0, NULL);
   root->cost = calculateCost(initial, final);
   pq.push(root);
   while (!pq.empty())
   {
      Node* min = pq.top();
      pq.pop();
      if (min->cost == 0)
      {
         printPath(min);
         return;
      }
      for (int i = 0; i < 4; i++)
      {
         if (isSafe(min->x + row[i], min->y + col[i]))
         {
            Node* child = newNode(min->mat, min->x, min->y, min->x + row[i],

   min->y + col[i],  min->level + 1, min);
            child->cost = calculateCost(child->mat, final);
           pq.push(child);
         }
      }
   }
}
int main()
{
   int initial[N][N] =
   {
      {1, 2, 3},
```

```
        {5, 6, 0},
        {7, 8, 4}
    };
    int final[N][N] =
    {
        {1, 2, 3},
        {5, 8, 6},
        {0, 7, 4}
    };
    int x = 1, y = 2;
    solve (initial, x, y, final);
    return 0;
}
```

Output-:

```
7 8 4

1 2 3
5 0 6
7 8 4

1 2 3
5 8 6
7 0 4

1 2 3
5 8 6
0 7 4
```

Practical:7

AIM-: Write a program to implement a heuristic search procedure in C++.

```
#include<bits/stdc++.h>
using namespace std;
vector<vector<int> > graph;
bool vis[100011];
int i,j;
vector<int> solve_vertex(int n,int e)
{
	vector<int> S;
	for(i=0;i<n;i++)
	{
		if(!vis[i])
		{
			for(j=0;j<(int)graph[i].size();j++)
			{
				if(!vis[graph[i][j]])
				{
					vis[i]=true;
					vis[graph[i][j]]=true;
					break;
				}
			}
		}
	}
	for(i=0;i<n;i++)
		if(vis[i])
			S.push_back(i);
	return S;
}
int main()
{
	int n,e,x,y;
	cout<<"Enter number of vertices:";
	cin>>n;
	cout<<"Enter number of Edges:";
	cin>>e;
	graph.resize(n);
	memset(vis,0,sizeof(vis));
	for(i=0;i<e;i++)
	{
		cout<<"Enter the end-points of edge "<<i+1<<" : ";
		cin>>x>>y;
		x--; y--;
		graph[x].push_back(y);
		graph[y].push_back(x);
	}
	vector<int> S = solve_vertex(n,e);
	cout<<"The required vertex cover is as follows:\n";
```

```
        for(i=0;i<(int)S.size();i++)
        cout<<S[i]+1<<" ";
        return 0;
}
```

Output: -

```
Enter number of vertices:3
Enter number of Edges:4
Enter the end-points of edge 1 : 1 2
Enter the end-points of edge 2 : 1 2
Enter the end-points of edge 3 : 3 1
Enter the end-points of edge 4 : 1 3
The required vertex cover is as follows:
1 2
```

Practical:8

AIM: -Write a program to implement a Production System in C++.

```
#include<iostream.h>
#include<conio.h>
int main()
{
char answer;
clrscr();
cout<<"Answer the following question to determine whether JOHN should get scholarship or
not?"<<endl;
cout<<"Q1) Is John a Student? (y/n)\n";
cin>>answer;
if(answer=='y') {
cout<<"John enjoys Student Life"<<endl;
cout<<"John Enjoys Student Life --> John Meets Friends"<<endl; cout<<"John Enjoys
Meets Friends --> John Needs Money"<<endl; cout<<"Q2) Does John has a job?(y/n)";
cin>>answer;
if(answer=='y')
{
cout<<"John Has Job --> John Has Free Time"<<endl;
cout<<"Since John Works in Free Time--> John Is Not Not Good In Studies "<<endl;
cout<<"John should not get the scholarship";
}
}
else
cout<<"John Will not recieve scholarship as he is not a student"<<endl;
return 0;
}
```

Output: -

```
Answer the following question to determine whether JOHN should get scholarship o
r not?
Q1) Is John a Student?(y/n)
y
John enjoys Student Life
John Enjoys Student Life --> John Meets Friends
John Enjoys Meets Friends --> John Needs Money
Q2) Does John has a job?(y/n)y
John Has Job --> John Has Free Time
Since John Works in Free Time--> John Is Not Not Good In Studies
John should not get the scholarship
```

Practical:9

AIM-: Write a program to implement An Expert System in C++.

```
#include <iostream>
using namespace std;
void measels(char,char,char,char,char);
void flu(char,char,char,char,char,char,char,char);
void cold(char,char,char,char,char); void chickenpox(char,char,char,char); int main()
{
char name[50];
char a,b,c,d,e,f,g,h,i,j,k;
cout << "Please enter your name.. " << endl;
cin>> name;
cout << "Do you have fever? (y/n)"<< endl;
cin>>a;
cout << "Do you have rashes? (y/n)"<< endl;
cin>>b;
cout << "Do you have headache? (y/n)"<< endl;
cin>>c;
cout << "Do you have running nose? (y/n)"<<
endl; cin>>d;
cout << "Do you have conjunctivities? (y/n)"<<
endl; cin>>e;
cout << "Do you have cough? (y/n)"<< endl;
cin>>f;
cout << "Do you have ache? (y/n)"<< endl;
cin>>g;
cout << "Do you have chills? (y/n)"<< endl;
cin>>h;
cout << "Do you have swollen glands? (y/n)"<<
endl; cin>>i;
cout << "Do you have snezzing? (y/n)"<< endl;
cin>>j;
cout << "Do you have sore throat? (y/n)"<< endl;
cin>>k;
measels(a,f,e,d,b); flu(a,c,g,e,h,k,f,d);
cold(c,j,k,d,h);
chickenpox(a,h,g,b);
return 0;
} void measels(char q,char w,char r,char t,char y)
{
if(q=='y'&&w=='y'&& r=='y' && t=='y' && y==
'y') cout<< "You may have measels."<< endl;
else cout<< "";
}
void flu(char q,char w,char r,char t,char y,char p,char l,char x)
{
if(q=='y'&&w=='y'&& r=='y' && t=='y' && y== 'y'&& p=='y' && l=='y' &&
x=='y') cout<< "You may have flu."<< endl;
```

else cout<< "";}

Output-:

```
--------------------Expert System: MEDICAL FIELD PROGRAM.----------------------


--------------------------DISEASES INCLUDED IN PROGRAM-------------------------
-----------------------------------[Stroke]-----------------------------------
-----------------------------------[FLU]--------------------------------------
-----------------------------------[COLD]-------------------------------------

please enter patient name:

Pariva
Does have the following symptom: Headache
 headache
Does have the following symptom: Runny nose
yes
Does have the following symptom: Sore throat
no
Does have the following symptom: Cough
yes
Does have the following symptom: Congestion
no
Does have the following symptom: Body Ache or Mild Headache Sneezing
no
Does have the following symptom: Fever
yes
```

Practical:10

AIM-: Write a program to implement A*Algorithm in C++.

```
#include <iostream>
void swap (int *p1, int *p2)
{
int temp = *p1;
*p1 = *p2;
*p2 = temp;
}
void bSort(int arrnumbers[], int n)
{
int i, j;
bool check;
for (i = 0; i < n-1; i++)
{
check = false;
for (j = 0; j < n-i-1; j++)
{
if (arrnumbers[j] > arrnumbers[j+1])
{
swap(&arrnumbers[j], &arrnumbers[j+1]);
check = true;
}}
if (check == false)
break;
}}
void print(int arrnumbers[], int sizeofarray)
{
int i;
for (i=0; i < sizeofarray; i++)
printf("%d ", arrnumbers[i]);
}
int main()
{
int arrnumbers[] = {5, 6, 1, 0, 2, 9};
int n = sizeof(arrnumbers)/sizeof(arrnumbers[0]);
bSort(arrnumbers, n);
printf("Sorted array: \n");
print(arrnumbers, n);
return 0;
}
```

Output-:

```
Sorted array:
0 1 2 5 6 9 Sorted array:
0 1 2 5 6 9
```