**PRACTICAL-10**

**AIM:**

Implement Deterministic Finite Automata.

**PROGRAM CODE:**

/\*

 \* DFA Simulation in C

 \*/

#include <stdio.h>

#include <stdlib.h>

struct node{

  int id\_num;

  int st\_val;

  struct node \*link0;

  struct node \*link1;

};

struct node \*start, \*q, \*ptr;

int vst\_arr[100], a[10];

int main(){

  int count, i, posi, j;

  char n[10];

  //clrscr();

  printf("\nPARTH PATEL\n19DCS098\n");

  printf("=-=-=-=-=-=-=-=-==-=-=-=-=-=-=-=-=-=-=-=\n");

  printf("Enter the number of states in the m/c:");

  scanf("%d",&count);

  q=(struct node \*)malloc(sizeof(struct node)\*count);

  for(i=0;i<count;i++)

  {

    (q+i)->id\_num=i;

    printf("State Machine::%d\n",i);

    printf("Next State if i/p is 0:");

    scanf("%d",&posi);

    (q+i)->link0=(q+posi);

    printf("Next State if i/p is 1:");

    scanf("%d",&posi);

    (q+i)->link1=(q+posi);

    printf("Is the state final state(0/1)?");

    scanf("%d",&(q+i)->st\_val);

  }

  printf("Enter the Initial State of the m/c:");

  scanf("%d",&posi);

  start=q+posi;

  printf("=-=-=-=-=-=-=-=-==-=-=-=-=-=-=-=-=-=-=-=\n");

  while(1){

    printf("=-=-=-=-=-=-=-=-==-=-=-=-=-=-=-=-=-=-=-=\n");

    printf("Perform String Check(0/1):");

    scanf("%d",&j);

    if(j){

      ptr=start;

      printf("Enter the string of inputs:");

      scanf("%s",n);

      posi=0;

      while(n[posi]!='\0'){

    a[posi]=(n[posi]-'0');

    //printf("%c\n",n[posi]);

    //printf("%d",a[posi]);

    posi++;

      }

      i=0;

      printf("The visited States of the m/c are:");

      do{

    vst\_arr[i]=ptr->id\_num;

    if(a[i]==0){

      ptr=ptr->link0;

    }

    else if(a[i]==1){

      ptr=ptr->link1;

    }

    else{

      printf("iNCORRECT iNPUT\n");

      return;

    }

    printf("[%d]",vst\_arr[i]);

    i++;

      }while(i<posi);

      printf("\n");

      printf("Present State:%d\n",ptr->id\_num);

      printf("String Status:: ");

      if(ptr->st\_val==1)

    printf("String Accepted\n");

      else

    printf("String Not Accepted\n");

    }

    else

      return 0;

  }

    printf("=-=-=-=-=-=-=-=-==-=-=-=-=-=-=-=-=-=-=-=\n");

  return 0;

}

**OUTPUT:**

