

NAME:- Sourav Paul

GROUP:- D2

REG. NO:- 20214056

BRANCH:- CSE DEPT.

ASSIGNMENT - 08

Q1. Implement Mealey and Moore machine in C.

Ans:-

- Mealey Machine implementation

```
#include<stdio.h>
#include<string.h>
void Mealey(char str[]){
    int n=0;
    int rem=0;
    while(str[n]!='\0'){
        if(str[n]=='0' || str[n]=='3'){
            printf("%d",rem);
        }
        else if(str[n]=='1'){
            switch(rem){
                case 0: rem=1;
                        printf("%d",rem);break;
                case 1: rem=2;
                        printf("%d",rem);break;
                case 2: rem=0;
                        printf("%d",rem);break;
            }
        }
        else if(str[n]=='2'){
            switch(rem){
                case 0: rem=2;
                        printf("%d",rem);break;
                case 1: rem=0;
                        printf("%d",rem);break;
                case 2: rem=1;
                        printf("%d",rem);break;
            }
        }
        n++;
    }
    printf("\n");
}
```

- Moore Machine Representation

```
void Moore(char str[]){
    int n=0;
    int rem=0;
    printf("%d",rem);
    while(str[n]!='\0'){
        if(str[n]=='0' || str[n]=='3'){
            printf("%d",rem);
        }
        else if(str[n]=='1'){
            switch(rem){
                case 0: rem=1;
                        printf("%d",rem);break;
                case 1: rem=2;
                        printf("%d",rem);break;
                case 2: rem=0;
                        printf("%d",rem);break;
            }
        }
        else if(str[n]=='2'){
            switch(rem){
                case 0: rem=2;
                        printf("%d",rem);break;
                case 1: rem=0;
                        printf("%d",rem);break;
                case 2: rem=1;
                        printf("%d",rem);break;
            }
        }
        n++;
    }
    printf("\n");
}

int main(){
    printf("input(in base 4):");
    char str[20];
    scanf("%s",str);
    printf("Mealey Machine output:");
    Mealey(str);
    printf("Moore Machine output:");
    Moore(str);
    return 0;
}
```

2)Write a C program to construct a NFA for the given regular expressions:

a) $(a+b)^*abb$

b) $0(0+1)^*0+1(0+1)^*+0+1$ c) $(0+1)^*(01+10)$

c) $0(0+1)*0+1(0+1)*+0+1$ c) $(0+1)*(01+10)$

d) $11(0+1)*11+111$

e) $((b*ab*ab*ab*)+b*)ab*$

```
#include<stdio.h>
#include<string.h>
#include<stdbool.h>

bool endsIn_abb(char str[]){
    int len=strlen(str);
    if(str[len-3]=='a' && str[len-2]=='b' && str[len-1]=='b'){
        return (true);
    }
    return (false);
}

bool startsAndEnds_0(char str[]){
    int len=strlen(str);
    if(str[0]=='0' && str[len-1]=='0'){
        return (true);
    }
    return (false);
}

bool startsWith_1(char str[]){
    if(str[0]=='1'){
        return (true);
    }
    return (false);
}

bool endsIn_01_or_10(char str[]){
    int len=strlen(str);
    if(str[len-2]=='0' && str[len-1]=='1'){
        return (true);
    }
    else if(str[len-2]=='1' && str[len-1]=='0'){
        return (true);
    }
    return (false);
}
```

```

bool startsAndEndsIn_11(char str[]){
    int len=strlen(str);
    if(len>2){
        if(str[0]=='1' && str[1]=='1'){
            if(str[len-2]=='1' && str[len-1]=='1'){
                return (true);
            }
            return (false);
        }
    }
    return (false);
}

```

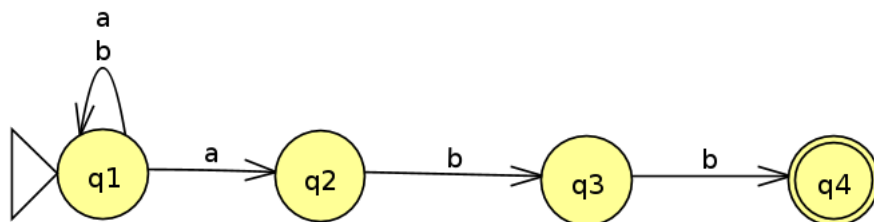
```

bool has_3plus1_a(char str[]){
    int len=strlen(str);
    int n=0, count=0;
    while(n<len){
        if(str[n++]=='a'){
            count++;
        }
    }
    if((count-1)%3==0){
        return (true);
    }
    return (false);
}

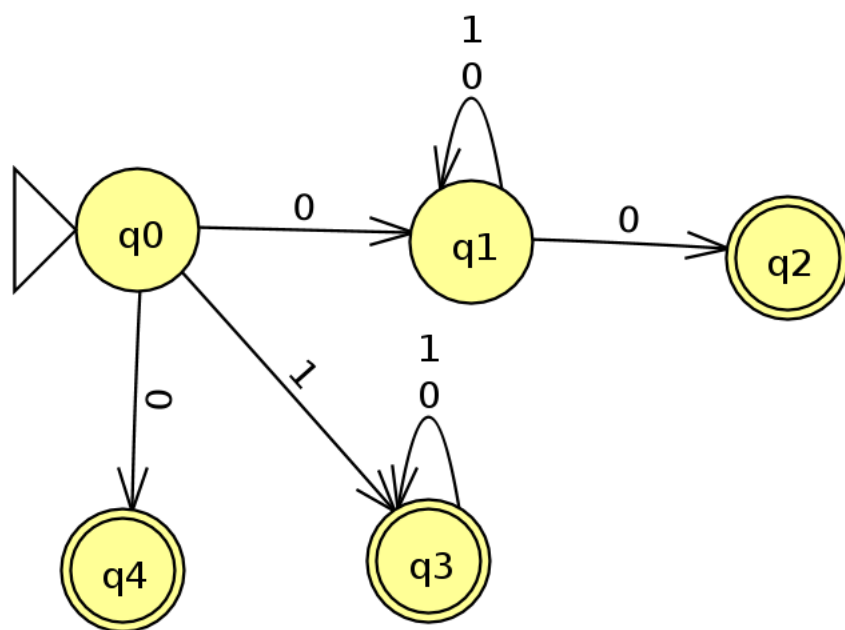
int main(){
    printf("input:");
    char str[20];
    scanf("%s",str);
    if(endsIn_abb(str)){
        printf("part of (a+b)*abb\n");
    }
    else printf("not part of (a+b)*abb\n");
    if(startsAndEnds_0(str) || startsWith_1(str)){
        printf("part of 0(0+1)*0+1(0+1)*+0+1\n");
    }
    else printf("not part of 0(0+1)*0+1(0+1)*+0+1\n");
    if(endsIn_01_or_10(str)){
        printf("part of (0+1)*(01+10)\n");
    }
    else printf("not part of (0+1)*(01+10)\n");
    if(startsAndEndsIn_11(str)){
        printf("part of 11(0+1)*11+111\n");
    }
    else printf("not part of 11(0+1)*11+111\n");
    if(has_3plus1_a(str)){
        printf("part of ((b*ab*ab*ab*ab)+b*)ab*\n");
    }
    else printf("not part of ((b*ab*ab*ab*ab)+b*)ab*\n");
    return 0;
}

```

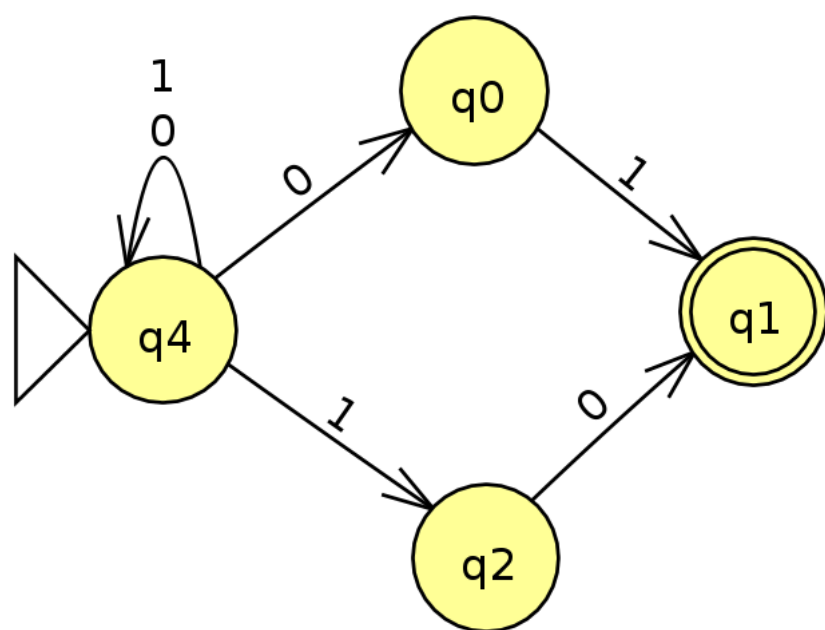
a)



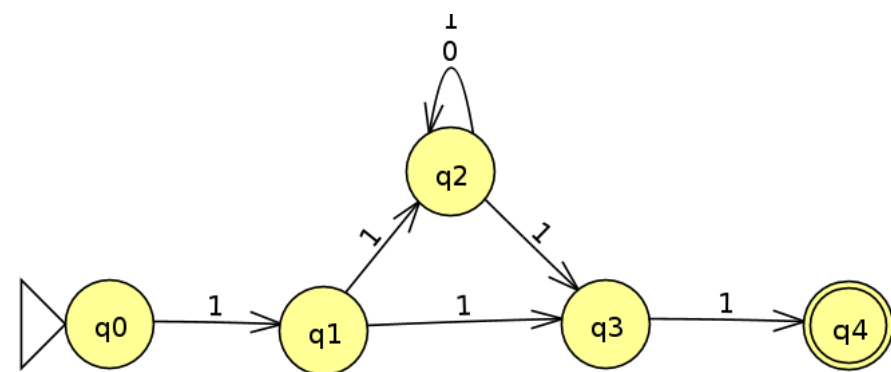
b)



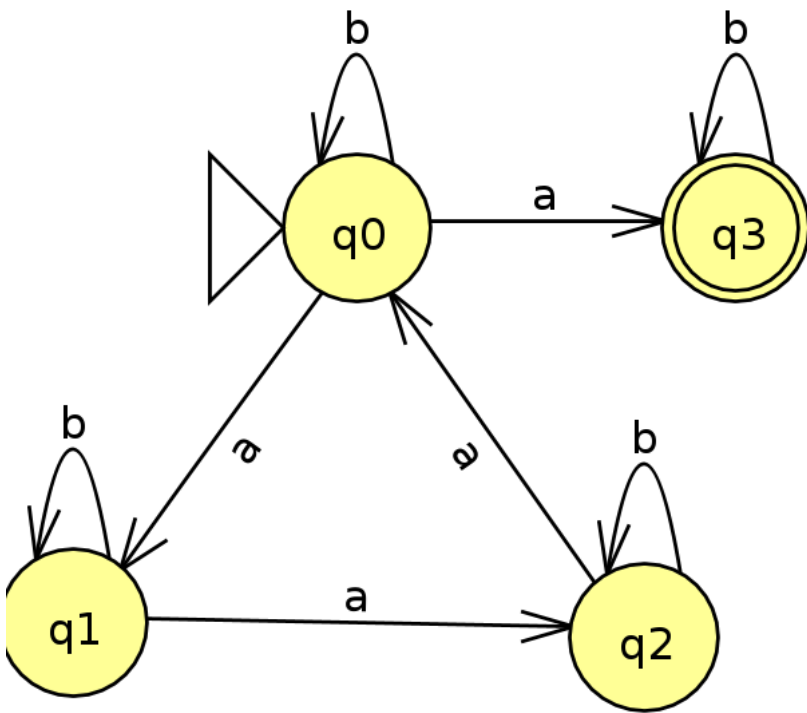
c)



d)



e)



3) Write a C program to check whether a given grammar is regular or not

```

#include<stdio.h>
#include<string.h>
#define s 0
int main(){
    int nt,t,p;
    printf("enter no. of non terminals, terminals, productions:");
    scanf("%d%d%d",&nt,&t,&p);
    char NT[nt], T[t];
    char P[p][20];
    printf("enter non Terminals:\n");
    for(int i=0;i<nt;i++){
        printf("\t");
        scanf("%s",NT);
    }
    printf("enter terminals:\n");
    for(int i=0;i<t;i++){
        printf("\t");
        scanf("%s",T);
    }
    printf("enter productions(format 'S=011S'):\n");
    for(int i=0;i<p;i++){
        printf("\t");
        scanf("%s[^\n]",P[i]);
    }
    for(int i=0;i<p;i++){
        int n=2;

        int len=strlen(P[i]);
        for(int j=0;j<nt;j++){
            while(n<len){
                if(P[i][n]==NT[j]){
                    if(P[i][n]!='\0'){
                        printf("Not a regular language!!\n");
                        return 0;
                    }
                }
                n++;
            }
        }
    }
    printf("regular language\n");
    return 0;
}

```