Program 3:

Design, Develop and Implement a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX)

a. Push an Element on to Stack
b. Pop an Element from Stack
c. Demonstrate how Stack can be used to check Palindrome
d. Demonstrate Overflow and Underflow situations on Stack
e. Display the status of Stack
f. Exit. Support the program with appropriate functions for each of the above operations

```
#include<stdio.h>
#define MAX 5
int s[MAX],top=-1,ele,i;

void push(int ele)
{
 if(top= =MAX-1)
  {
   printf("Stack Over flow");
   return;
  }
            top++;
            s[top]=ele;
}

int pop()
   {
            int ele;
            if(top= =-1)
            {
             printf("Stack underflow");
             return;
             }
            ele=s[top];
            top--;
            return ele;
   }

void display()
{
if(top==-1)
            {
             printf("Stack underflow");
            return;
             }
printf("Stack Contents are\n");
            for(i=top;i>=0;i--)
```

```
        printf("%d\n",s[i]);
}


void pal()
{
top=-1;
int i=1,len=0,rev=0,digit,temp,n;
printf("Enter a Number\n");
scanf("%d",&n);

temp=n;
while(n!=0)
{
        digit=n%10;
        n=n/10;
        push(digit);
        len++;
}
while(len!=0)
{
        digit=pop();
        rev=rev+(digit*i);
        len--;
        i=i*10;
}
if(temp==rev)
printf("Number is a palindrome");
else
printf("Number is
not a palindrome");
}

void main()
{
int ch;
do
{
printf(“1:push    2:pop      3:display      4:palindrome\n”);
printf("Enter your choice\n");
scanf("%d",&ch);

switch(ch)
{
case 1:
        printf("Enter the element to be pushed \n");
        scanf("%d",&ele);
```

```
            push(ele);
            break;
case 2:
            ele=pop();
            printf("Element deleted is %d",ele);
            break;
case 3:
            display();
            break;
case 4:
            pal();
            break;
default: printf("invalid choice\n");
}
}while(ch<=4);
}
```

Output

```
1:push    2:pop      3:display      4:palindrome
Enter your choice
1
Enter the element to be pushed
10
Enter your choice
1
Enter the element to be pushed
20
Enter your choice
1
Enter the element to be pushed
30
Enter your choice
1
Enter the element to be pushed
40
Enter your choice
1
Enter the element to be pushed
50
Enter your choice
1
Enter the element to be pushed
60
Stack overflow
Enter your choice
3
Stack Contents are
```

```
50
40
30
20
10

Enter your choice
2
Element deleted is 50
Enter your choice
3
Stack Contents are
40
30
20
10
Enter your choice
2
Element deleted is 40
Enter your choice
2
Element deleted is 30
Enter your choice
2
Element deleted is 20
Enter your choice
2
Element deleted is 10
Enter your choice
2
Stack Underflow
Enter your choice
4
Enter a Number
1234
Number is not a Palindrome
Enter your choice
4
Enter a Number
1221
Number is  a Palindrome
Enter your choice
5
```

Program 4:
Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, - , *, /, %(Remainder), ^(Power) and alphanumeric operands

```
#include<stdio.h>
#include<process.h>

char infix[25],postfix[25];
int s[10],top=-1;
void push(char c);
int pri(char c);
char pop();

int pri(char c)
{
switch(c)
{
case  '+':
case  '-': return 1;
case  '*':
case  '/': return 2;
case  '^': return 3;
case  '(':
case  '#': return 0;
}
}

void push(char c)
{
top++;
s[top]=c;
}


char pop()
{
char e;
e=s[top];
top--;
return e;
}

void main()
{
int i,j=0;
char ch,ele;
```

```
printf("enter the infix expression\n");
scanf("%s",infix);

push('#');

for(i=0;infix[i]!='\0';i++)
{
ch=infix[i];
if(isalnum(ch))
postfix[j++]=ch;
else if(ch=='(')
push(ch);
else if(ch==')')
{
while(s[top]!='(')
postfix[j++]=pop();
ele=pop();
}
else
{
while(pri (s[top]) >= pri (ch))
postfix[j++]=pop();
push(ch);
}
}
while(s[top]!='#')
postfix[j++]=pop();
postfix[j]='\0';
printf("postfix expression is %s",postfix);
getch();
}
```

OUTPUT

```
enter the infix expression
2+3*5
postfix expression is 235*+



enter the infix expression
(2+3)*5
postfix expression is 23+5*
```

Program 5:

Design, Develop and Implement a Program in C for the following Stack applications
a. Evaluation of Suffix expression with single digit operands and operators: +, - , *, /, %, ^
b. Solving Tower of Hanoi problem with n disks

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
#include<math.h>

int op1,op2,res,i,top=-1,s[10],ele,n;

 void push(int ele)
  {
  top++;
  s[top]=ele;
  }

  int pop()
  {
  int ele;
  ele=s[top];
  top--;
  return(ele);
  }


void main()
{
int e;
char postfix[20],ch;

printf("enter the postfix exp\n");
scanf("%s",postfix);

for(i=0;postfix[i]!='\0';i++)
{
ch=postfix[i];


if(isdigit(ch))
push(ch-'0');
else
{
op2=pop();
op1=pop();

switch(ch)
{
case '+':res=op1+op2;
                  break;
case '-':res=op1-op2;
                 break;
case '*':res=op1*op2;
              break;
case '/':res=op1/op2;
              break;
case '^':res=pow(op1,op2);
               break;
 }
push(res);
 }
 }
  printf("result of postfix exp %d",res);
}
```

OUTPUT

```
enter the postfix exp
23+5*
result of postfix exp is 25

enter the postfix exp
235*+
result of postfix exp is 17
```

Tower of Hanoi

```
#include<stdio.h>

void tow(int n, char s, char t, char d)
{
if(n= =1)
{
printf("move disk 1 from %c to %c\n",s,d);
return;
}

tow(n-1,s,d,t);
printf("move disk %d from %c to %c\n", n, s, d);
tow(n-1,t,s,d);
}
void main()
{
        int n;
        printf("enter number of disks\n");
        scanf("%d",&n);

        tow(n,'s','t','d');
}
```

OUTPUT

```
Enter the number of disks : 1
Move disk 1 from  to s to d

Enter the number of disks : 3
Move disk 1 from  to s to d
Move disk 2 from s to t
Move disk 1 from d to t
Move disk 3 from s to d
Move disk 1 from t to s
Move disk 2 from t to d
Move disk 1 from s to d
```

Program 6:

Design, Develop and Implement a menu driven Program in C for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX)

a. Insert an Element on to Circular QUEUE

b. Delete an Element from Circular QUEUE

c. Demonstrate Overflow and Underflow situations on Circular QUEUE

d. Display the status of Circular QUEUE

e. Exit

Support the program with appropriate functions for each of the above operations

```
#include<stdio.h>
#define MAX 3
int cq[MAX],f=0,r=-1,i,j,c=0,n,ele;

void insert()
{
if(c==MAX)
{
printf("CIRCULAR QUEUE OVERFLOW|n");
return;
}
printf("enter the value\n");
scanf("%d",&ele);
r=(r+1)%MAX;
cq[r]=ele;
c++;
}

void delete_ele()
{
if(c==0)
{
printf("CIRCULAR QUEUE UNDERFLOW|n");
f=0,r=-1;
return;
}
printf("element deleted is %d",cq[f]);
f=(f+1)%MAX;
c--;
}


void display()
{
j=f;
if(c= =0)
{
printf("CIRCULAR QUEUE UNDERFLOW|n");
return;
}
printf("contents of CQ are\n");
for(i=1;i<=c;i++)
{
printf("%d\t", cq[j]);
j=(j+1)%MAX;
}
}
void main()
{
int ch;
do
{
printf("1:insert 2:delete 3:display \n");
printf("enter your choice\n”);
scanf("%d",&ch);
switch(ch)
{
case 1:insert();
            break;
case 2:delete_ele();
            break;
case 3:display();
            break;
default:printf("invalid choice\n");
}
}while(ch<=3);
}
```

Output:

```
1:insert 2:delete 3:display
enter your choice
1
enter the value
10
1:insert 2:delete 3:display
enter your choice
1
enter the value
20
1:insert 2:delete 3:display
```

```
enter your choice
1
enter the value
30
1:insert 2:delete 3:display
enter your choice
1
CIRCULAR QUEUE OVERFLOW
1:insert 2:delete 3:display
enter your choice
3
contents of CQ are
10      20      30
1:insert 2:delete 3:display
enter your choice
2
element deleted is 10
1:insert 2:delete 3:display
enter your choice
3
contents of CQ are
20      30
1:insert 2:delete 3:display
enter your choice
1
enter the value
40
1:insert 2:delete 3:display
enter your choice
3
contents of CQ are
20      30      40
1:insert 2:delete 3:display
enter your choice
1
CIRCULAR QUEUE OVERFLOW
```