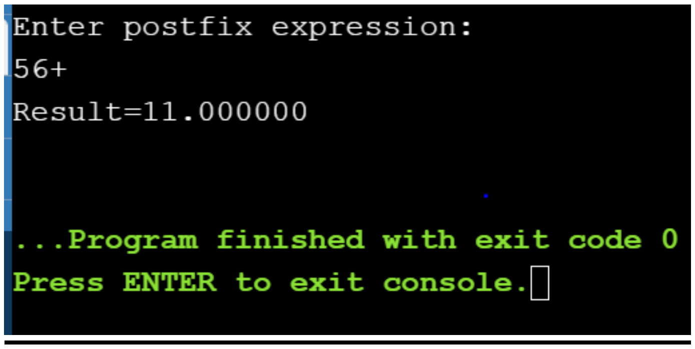
Q1

CODE:

```
#include <stdio.h>

#include<string.h>

#include<math.h>

double compute(char symbol,double op1,double op2)

{

        switch(symbol)

        {

                case '+':return op1+op2;

                case '-':return op1-op2;

                case '*':return op1*op2;

                case '/':return op1/op2;

                case '$':

                case '^':return pow(op1,op2);

        }

}

void main()

{

        double s[20];

        double op1,op2,res;

        int top,i;

        char postfix[20],symbol;

        top=-1;

        printf("Enter postfix expression:\n");
```

```
    scanf("%s",&postfix);
    for(i=0;i<strlen(postfix);i++)
    {
        symbol=postfix[i];
        if(isdigit(symbol))
        {
            s[++top]=symbol-'0';
        }
        else
        {
            op2=s[top--];
            op1=s[top--];
            res=compute(symbol,op1,op2);
            s[++top]=res;
        }
    }
    res=s[top--];
    printf("Result=%f\n",res);
}
```

OUTPUT:



```
Enter postfix expression:
56+
Result=11.000000


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

Q2

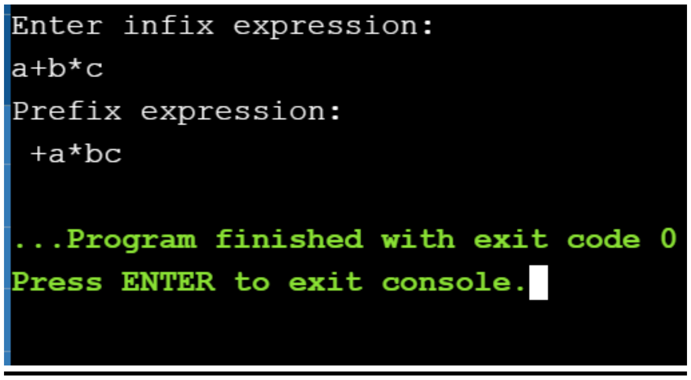
CODE:

```
#include <stdio.h>

#include<string.h>

void reverse(char *s)

{

    int i=0,j=0;

    int len=strlen(s);

    char temp[20];

    for(i=len-1;i>=0;i++)

    {

        temp[j]=s[i];

        j++;


    }

    strcpy(s,temp);

}


int F(char symbol)

{

    switch(symbol)

    {

        case '+':

        case '-':return 1;
```

```
                case '/':
                case '*':return 3;
                case '^':
                case '$':return 6;
                case ')':return 0;
                case '#':return -1;
                default:return 8;
        }
}
int G(char symbol)
{
        switch(symbol)
        {
                case '+':
                case '-':return 2;
                case '/':
                case '*':return 4;
                case '^':
                case '$':return 5;
                case '(':return 0;
                case ')':return 9;
                default:return 7;
        }
}
void infix_prefix(char infix[],char prefix[])
{
        int top=-1,j=0,i;
        char s[30];
        char symbol;
        s[++top]='#';
        reverse(infix);
        for(i=0;i<strlen(infix);i++)
        {
                symbol=infix[i];
                while(F(s[top])>G(symbol))
                {
                        prefix[j]=s[top--];
                        j++;
                }
                if(F(s[top])!=symbol)
                        s[++top]=symbol;
                else
                        top--;
        }
        while(s[top]!='#')
        {
                prefix[j++]=s[top--];
        }
        prefix[j]='\0';
        reverse(prefix);
        printf("Prefix expression:\n %s",prefix);
}
```

```
void main()

{

        char infix[20];

        char prefix[20];

        printf("Enter infix expression:\n");

        scanf("%s",&infix);

        infix_prefix(infix,prefix);


}
```

OUTPUT:



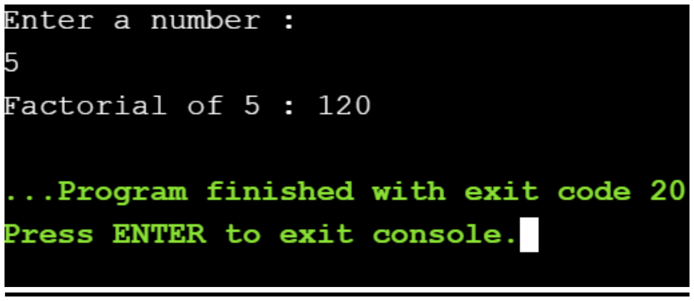
```
Enter infix expression:
a+b*c
Prefix expression:
 +a*bc

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

Q2

CODE:

```
#include <stdio.h>

int fact(int n)

{

        if(n==0)return 1;
        return n*fact(n-1);
}
void main()
{
        int n;
        printf("Enter a number :\n");
        scanf("%d",&n);
        printf("Factorial of %d : %d",n,fact(n));
}
```

OUTPUT:



```
Enter a number :
5
Factorial of 5 : 120

...Program finished with exit code 20
Press ENTER to exit console.
```