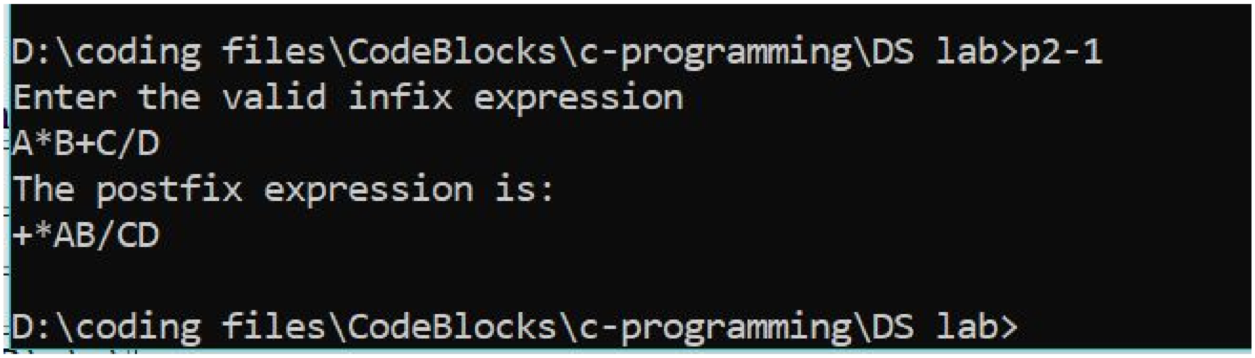
PRACTICE-2

1--

```
/*WAP to convert a given valid parenthesized infix arithmetic expression to prefix
expression. The expression consists of single character operands and the binary operators
+ (plus), - (minus), * (multiply) and / (divide).
*/

#include<stdio.h>
#include<string.h>
int F(char symbol)
{
        switch(symbol)
        {
        case '+':
        case '-': return 1;
        case '*':
        case '/':return 3;
        case '^':
        case '$':return 6;
        case '(':return 9;
        case'#':return 0;
        default:return 8;
}
}
int G(char symbol)
{
        switch(symbol)
        {
        case '+':
        case '-': return 2;
        case '*':
        case '/':return 4;
        case '^':
        case '$':return 5;
        case '(':return 0;
        case')':return 9;
        default:return 7;
}
}
void infix_prefix(char infix[],char prefix[])
{
        int top,i,j;
        char s[30],symbol;
        top=-1;
        s[++top]='#';
        j=0;
  strrev(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])!=G(symbol))
                s[++top]=symbol;
                else
                top--;
        }
        while(s[top]!='#')
        {
                prefix[j++]=s[top--];
        }
        prefix[j]='\0';
  strrev(prefix);
}
int main()
{
        char infix[20];
        char prefix[20];
        printf("Enter the valid infix expression\n");
        scanf("%s",infix);
        infix_prefix(infix,prefix);
        printf("The prefix expression is:\n");
        printf("%s\n",prefix);
}
```

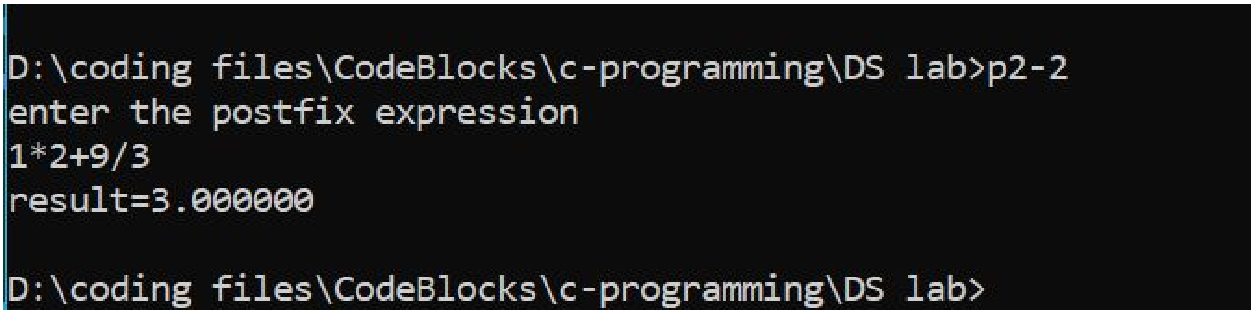


```
D:\coding files\CodeBlocks\c-programming\DS lab>p2-1
Enter the valid infix expression
A*B+C/D
The postfix expression is:
+*AB/CD

D:\coding files\CodeBlocks\c-programming\DS lab>
```

2--

/*WAP to demonstrate the Evaluation of postfix expression.*/

```
#include<stdio.h>
#include<math.h>
#include<string.h>
#include<ctype.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;
  float res;
  int top,i;
  char postfix[20],symbol;
  printf("enter the postfix expression \n");
  scanf("%s",postfix);
  top=-1;
  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);
  }
```



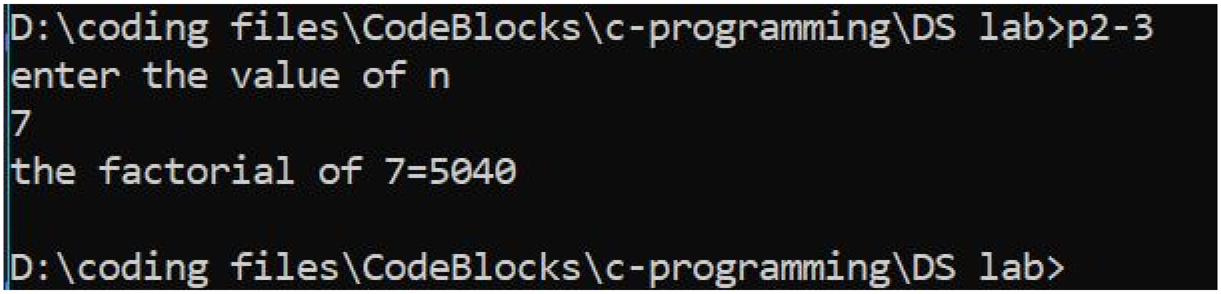
```
D:\coding files\CodeBlocks\c-programming\DS lab>p2-2
enter the postfix expression
1*2+9/3
result=3.000000

D:\coding files\CodeBlocks\c-programming\DS lab>
```

3--

/*WAP to perform factorial of a number using Recursion*/

```
#include <stdio.h>
int fact(int n)
{
   if(n==0)
   return 1;
   else
   return n*fact(n-1);
}
void main()
{
   int n;
```

```
    printf("enter the value of n \n");
    scanf("%d",&n);
    printf("the factorial of %d=%d\n",n,fact(n));
}
```

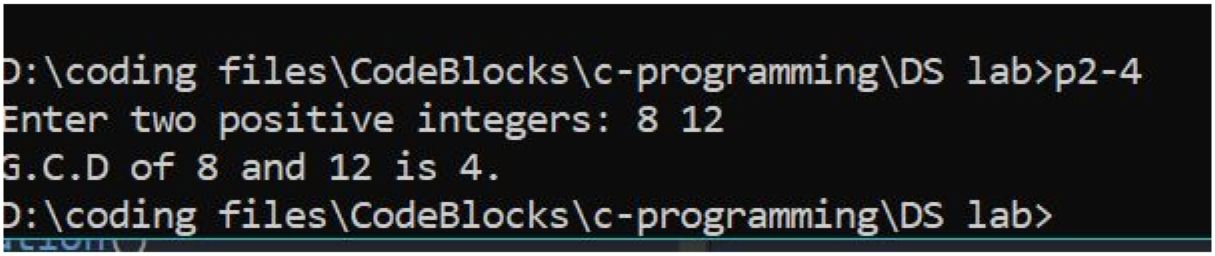


```
D:\coding files\CodeBlocks\c-programming\DS lab>p2-3
enter the value of n
7
the factorial of 7=5040

D:\coding files\CodeBlocks\c-programming\DS lab>
```

4--

/*WAP to perform GCD of two numbers using Recursion.*/

```
#include <stdio.h>
int gcd(int n, int m);
int main() {
    int n, m;
    printf("Enter two positive integers: ");
    scanf("%d %d", &n, &m);
    printf("G.C.D of %d and %d is %d.", n, m, gcd(n, m));
    return 0;
}
int gcd(int n, int m) {
    if (m != 0)
        return gcd(m, n % m);
    else
        return n;
}
```



```
D:\coding files\CodeBlocks\c-programming\DS lab>p2-4
Enter two positive integers: 8 12
G.C.D of 8 and 12 is 4.
D:\coding files\CodeBlocks\c-programming\DS lab>
```