Week - 3 Extra Programs

① C- Program to convert infix expression to Prefix expression:-

```
#include <stdio.h>
#include <string.h>
#include <Process.h>
 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 G1(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[30], Refix[30];
  Printf("Enter the valid infix expression:\n");
  scanf("%s", infix);
  infix_Prefix (infix, Prefix);
  Printf("the Prefix expression is :\n");
  Printf("%s\n", Prefix);
  return 0;
}
```

② C-Program to demonstrate the evaluation of Postfix expression.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.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 '^' : return Pow(op1,op2);
  } default : exit(0);
```

```
int main()
{
 double s[20];
 double op1, op2;
 double result;
  int top, i;
 char Postfix[20], symbol;
 Printf("\n Enter the valid 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--];
   result = compute (symbol, op1, op2);
    s[++top] =result;
  }
}

 result= s[top--];
 Printf("RESULT = %f\n", result);
 return 0;
}
```

③ C-Program to Perform factorial of a number using recursion:-

```
#include <stdio.h>
  int fact(int n)
{
   if (n==0)
   return 1;
   return n*fact(n-1);
}
int main()
{
   int n;
   Printf("Enter the value of n\n");
   scanf("%d",&n);
   Printf("The factorial of %d= %d \n", n,
   return 0;                          fact(n));
}
```

④ C-Program to Perform GCD of two numbers using recursion:-

```
#include <stdio.h>
 int gcd(int num1, int num2);
  int main()
{
   int num1, num2;
   Printf("\n Enter two Positive integers:\n");
   scanf("%d %d", &num1, &num2);
```

```
Printf("G.C.D of %d and %d is %d", num1, num2,
        gcd(num1, num2));
  return 0;
}

int gcd(int num1, int num2)
{
  if (num2 != 0)
    return gcd(num2, num1 % num2);
  else
    return num1;
}
```