WEEK 4 PRACTICE PROGRAMS

1.**Infix to prefix**

```
#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 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, j, i;
    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);
}

void main()
{
    char infix[30], prefix[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);

}
```

```
Enter the valid infix expression:
(A+(B-C)*D)
The prefix expression is:
+A*-BCD
```

2. **Evaluation of postfix**

```
#include <stdio.h>
#include <stdlib.h>
#include<string.h>
#include<math.h>
//evaluation of postfix
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);
        }
}
int main()
{
double s[20];
double res;
double op1,op2;
int top,i;
char postfix[20],symbol;
```

```
printf("Please enter the postfix expression");
scanf("%s",postfix);
top=-1;
for(int 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 is :%0.2f",res);
   return 0;
}
```

```
Please enter the postfix expression
123+*321-+*
result is :20.00
Process returned 0 (0x0)   execution time : 38.201 s
Press any key to continue.
```

3. **Factorial by Recursion**

```
#include <stdio.h>
#include <stdlib.h>
int fact(int n)
{
   return((n==0)?1:(n*fact(n-1)));
}
int main()
{
   int n;
```

```
   printf("Please enter the number whose factorial you need to find:");
   scanf("%d",&n);
   int res=fact(n);
   printf("%d",res);
    return 0;
}
```

```
Please enter the number whose factorial you need to find:5
120
Process returned 0 (0x0)   execution time : 2.704 s
Press any key to continue.
```

4. **GCD by Recursion**

```
#include <stdio.h>
#include <stdlib.h>
int gcd(int a,int b)
{
    return ((b==0)?a:gcd(b,a%b));
}
int main()
```

```
{
    int x,y;
    printf("Please enter the values a and b:");
    scanf("%d",&x);
    scanf("%d",&y);
    int res=gcd(x,y);
    printf("%d",res);
    return 0;
}
```