07/10/11

1. Program to evaluate prefix 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);
 }
}
```

```
int main()
{
  double s[20];
  double res;
  double op1, op2;
  int top, i;
  char prefix[20], symbol;
  printf("Enter the prefix expression:\n");
  scanf("%s", prefix);
  top = -1;
  strrev(prefix);
  for(i=0; i< strlen(prefix); i++)
  {
    symbol = prefix[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);
}
```

2. Program to convert valid infix expression to prefix.

```
#include<stdio.h>
#include<string.h>
int F (char symbol)
{
  switch (symbol)
  {
  case '+':
  case '-': return 1;
```

```
Case '*':
Case '/': return 3;
Case '^':
case '$': return 6;
Case ')': return 0;
Case '#': return -1;
default: return 8;
}
}
int G(char symbol)
{
    switch(symbol)
    {
     case '+':
     case '-': return 1;
     Case '*':
     Case '/': return 3;
     Case '^':
     Case '$': return 6;
     Case '(': return 0;
     Case ')': return 9;
     default: return 7;
    }
}
void infix_prefix(char infix[], char prefix[])
{
    int top;
    char s[30];
      int j;
      int i;
      char 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);
}
```

3.
```
#include <stdio.h>
int gcd (int n1, int n2);
  int main()
   {
   int n1, n2;
   printf ("Enter two number : \n");
   scanf ("%.d %.d", &n1, &n2);
   printf ("Gcd of two numbers is %.d", gcd (n1, n2));
      return 0;
  }
 int gcd (int n1, int n2)
  {
   if (n2 != 0)
    return gcd(n2, n1 %. n2);
      else
    return n1;
  }
```

4.
```
# include <stdio.h>
int fact (int n)
 {
   if (n == 0)
   return 1;
   else
   return n*fact(n-1);
 }
 int main()
{
   int n ;
   printf ("Enter the value of n\n");
   Scanf ("%.d",&n);
   printf ("The factorial of %.d is %.d\n", n, fact (n));
}
```