1). Postfix evaluation:-

```
#include <stdio.h>
#include <math.h>
#include <string.h>
double compute (char symbol, double op
 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 res, op1, op2;
int top, i;
char postfix[20], symbol;
printf("Enter 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 of the given postfix operation
    %f\n", res);
}
```

Program 2: infix to prefix:-

```
#include <stdio.h>
#include <string.h>
#include <stdlib.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))
    {
      ~~prefix~~ s[++top] = symbol;
    }
    else
    {
      top--;
    }
}
while (s[top] != G('#'))
{
    s[++top] = symbol;
}
```

```
    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);
}
```

3:- factorial :-

```
#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("The factorial of %d = %d", n, fact(n
}
```

4: GCD.

```
#include<stdio.h>
int GCD(int a, int b)
{
  if(b != 0)
  {
    if(a%b == 0)
      return b;
    else
      GCD(b, (a%b));
  }
  else
    printf("Invalid set of numbers.\n");
}
void main()
{
  int a, b, t;
```

```
printf("Enter 2 numbers \n");
scanf("%d %d", &a, &b);
if (a < b)
{
    t = a;
    a = b;
    b = t;
}
printf("GCD of %d and %d is %d",
        a, b, GCD(a,b));
}
```