Practice programs :-

```
#include<stdio.h>
#include<string.h>

int F (char symbol)
{
   switch (symbol)
   {
    case '+':
    case '-': return 1;
    case '*':
    case '/': return 3;
    case '^':
    case '$': return 6;
    case '(': return 0;
    case '#': return -1;
    default : return 8;
  }
int G (char symbol)
 {
   switch(symbol)
   {
    case '+':
    case '-': return 2;
    case '*':
    case '/': return 4;
    case '^':
    case '$': 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);
}
Void main() {
 char infix[20], prefix[20];
  printf("\n Enter the valid infix expression: ");
  scanf("%d", &infix);
  infix_prefix(infix, prefix);
  printf("The prefix expression is: %s\n", prefix);
}
```

```
Enter the valid infix Expression:(A+(B-C)*D)

The postfix expression is: +A*-BCD
```

```
#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], res, op1, op2;
  int top, i;
  char postfix[20], symbol;
  printf("Enter the valid postfix Expression:");
  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("\n The result is: %f", res);
}
```

```
Enter the valid postfix Expression: 123+*321-+*

The Result is: 20.000000
```

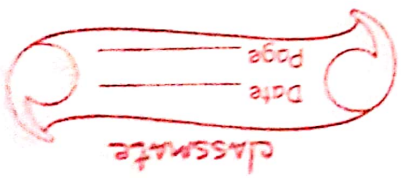
3) #include <stdio.h>

```
int fact (int n)
{
    if (n==0)
        return 1;
    return n* fact (n-1);
}
int main()
{
    int num, res;
    printf("\n Enter the number to find it's factorial: ");
    scanf ("%d", &num);
    res = fact(num)
    printf ("\n The factorial is : %d", res);
    return 0;
}
```

```
Enter the number to find its factorial: 7

The factorial is: 5040
```

4)
```
#include<stdio.h>
int GCD(int num1, int num2)
{
  if (num1> num2)
  {
    if (num2!=0)
      return GCD(num2, num1%num2);
    else
      return num1;
  }
  else {
    if (num1!=0)
      return GCD(num1, num2%num1);
    else
      return num2;
  }
}
```



```
int main()
{
    int num1, num2, res;
    printf("Enter the two numbers:");
    scanf("%d%d", &num1, &num2);
    res = GCD(num1, num2);
    printf("The Greatest common Divisor of %d & %d is: %d", num1, num2, res);
    return 0;
}
```

```
Enter the two numbers:  24 8
The Greatest Common Divisor of 24 & 8 is:  8
```