main.c

```c
# include <stdio.h>
# include <string.h>
# define MAX 20
void infixtoprefix(char infix[20], char prefix[20]);
void reverse(char array[30]);
char pop();
void push(char symbol);
int isOperator(char symbol);
int prcd(char symbol);
int top = -1;
char stack[MAX];

main() {
char infix[20], prefix[20], temp;
printf("Enter infix operation: ");
gets(infix);
infixtoprefix(infix, prefix);
reverse(prefix);
puts((prefix));
}
void infixtoprefix(char infix[20], char prefix[20]) {
int i, j = 0;
char symbol;
stack[++top] = '#';
reverse(infix);
for (i = 0; i < strlen(infix); i++) {
symbol = infix[i];
if (isOperator(symbol) == 0) {
  prefix[j] = symbol;
  j++;
} else {
  if (symbol == ')') {
    push(symbol);
  } else if (symbol == '(') {
    while (stack[top] != ')') {
      prefix[j] = pop();
      j++;
```

main.c

```
39      pop();
40    } else {
41      if (prcd(stack[top]) <= prcd(symbol)) {
42        push(symbol);
43      } else {
44        while (prcd(stack[top]) >= prcd(symbol)) {
45          prefix[j] = pop();
46          j++;
47        }
48        push(symbol);
49      }
50
51      }
52    }
53
54    }
55
56   while (stack[top] != '#') {
57     prefix[j] = pop();
58     j++;
59    }
60   prefix[j] = '\0';
61  }
62
63  void reverse(char array[30]) {
64
65  int i, j;
66  char temp[100];
67  for (i = strlen(array) - 1, j = 0; i + 1 != 0; --i, ++j) {
68   temp[j] = array[i];
69  }
70  temp[j] = '\0';
71  strcpy(array, temp);
72
73  }
74
75   char pop() {
```

ain.c

```
 char pop() {
 char a;
 a = stack[top];
 top--;
 return a;
 }

void push(char symbol) {
top++;
stack[top] = symbol;
}

int prcd(char symbol) {

switch (symbol) {
 case '+':
  case '-':
   return 2;
   break;
 case '*':
  case '/':
   return 4;
   break;
  case '$':
  case '^':
   return 6;
   break;
  case '#':
  case '(':
  case ')':
   return 1;
   break;
  }
}

int isOperator(char symbol) {
switch (symbol) {
```

main.c

```
    return 0;
    break;
   case '#':
   case '(':
   case ')':
    return 1;
    break;
  }
}

int isOperator(char symbol) {
switch (symbol) {
case '+':
case '-':
case '*':
case '/':
case '^':
case '$':
case '&':
case '(':
case ')':
  return 1;
  break;
default:
  return 0;

  }
 }
```

input

```
main.c:(.text+0x2e): warning: the `gets' function is dangerous and should not be used.
Enter infix operation: ((a-b/c)*(a/k-l))
*-a/bc-/akl


...Program finished with exit code 0
Press ENTER to exit console.
```

```c
#include<stdio.h>
#include<ctype.h>
#include<math.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 '$':
        case '^':return pow(op1,op2);

    }
}
void main()
{
double s[20];
double res;
double op1,op2;
int top,i;
char postfix[20],symbol;
printf("Enter the 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;
```

main.c

```
    }
}
void main()
{
double s[20];
double res;
double op1,op2;
int top,i;
char postfix[20],symbol;
printf("Enter the 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=%f\n",res);
}
```

input

```
Enter the postfix expression
53+62/*35*+
result=39.000000


...Program finished with exit code 0
Press ENTER to exit console.
```

main.c

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

input

```
Enter the value of n
5
The factorial of 5 =120


...Program finished with exit code 24
Press ENTER to exit console.
```

main.c

```c
#include <stdio.h>
int GCD(int, int);
int main()
{
int num1, num2, res;
printf("\n Enter the two numbers: ");
scanf("%d %d", &num1, &num2);
res = GCD(num1, num2);
printf("\n GCD of %d and %d = %d", num1, num2, res);
return 0;
}
int GCD(int x, int y)
{
int rem;
rem = x%y;
if(rem==0)
 return y;
else
 return (GCD(y, rem));
}
```

input

```
 Enter the two numbers: 80 56

 GCD of 80 and 56 = 8

...Program finished with exit code 0
Press ENTER to exit console.
```