

## ***DS LAB-PRACTICE PROGRAMS (4 qns)***

### ***Programs and Output***

***Mallika Prasad***

***1BM19CS081***

***7.10.2020***

#### ***Program 2-***

```
#include <stdio.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);
    }
}
int main()
{
    double s[20];
    double res;
    double op1, op2;
```

```
int top, i;
char postfix[20], symbol;
printf("enter postfix exp:\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 is %f\n",res);
return 0;
}
```

***Output-***

```
main.c
36     symbol-postfix[i];
37     if(isdigit(symbol))
38     s[++top] = symbol-'0';
39     else
40     {
41         op2=s[top--];
42         op1=s[top--];
43         res=compute(symbol,op1,op2);
44         s[++top]=res;
45     }
46 }
47 res=s[top--];
48 printf("result is %f\n",res);
49 return 0;
50 }
51

input
main.c:37:16: warning: implicit declaration of function 'isdigit' [-Wimplicit-function-declaration]
enter postfix exp:
123**321--+
result is 20.000000

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

### Program 3-

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

```
int fact(int n)
```

```
{
```

```
    if(n==0)
```

```
        return 1;
```

```
        return n*fact(n-1);
```

```
}
```

```
int main()
```

```
{
```

```
    int n;
```

```
    printf("Enter the value of n\n");
```

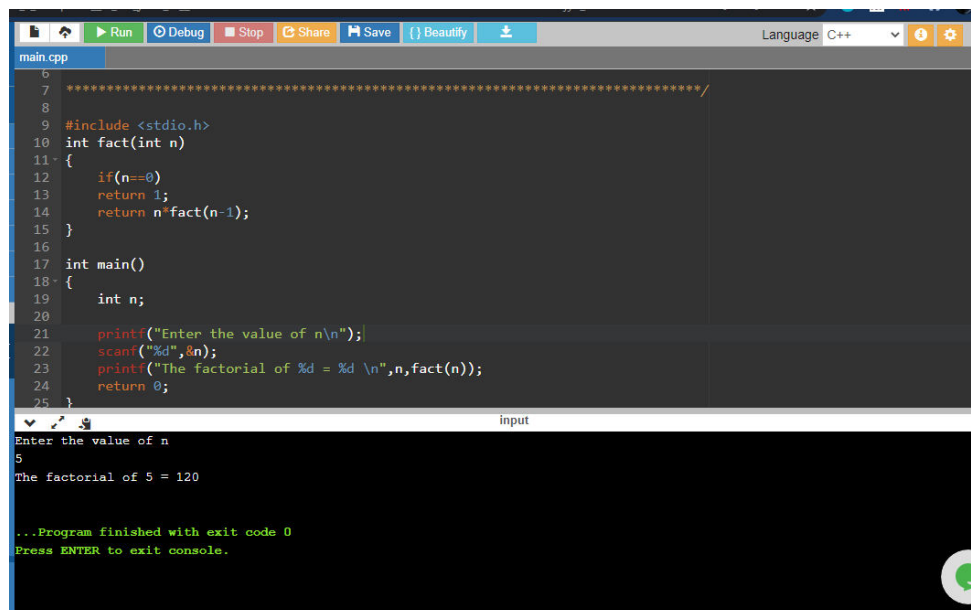
```
    scanf("%d",&n);
```

```
    printf("The factorial of %d = %d \n",n,fact(n));
```

```
    return 0;
```

```
}
```

## Output-



The screenshot shows a C++ IDE with a file named 'main.cpp'. The code defines a recursive function 'fact' to calculate the factorial of a number 'n'. The 'main' function prompts the user to enter a value for 'n', reads the input, and prints the factorial result. The output window shows the user entering '5' and the program outputting 'The factorial of 5 = 120'. The program finishes with exit code 0.

```
main.cpp
6
7 *****
8
9 #include <stdio.h>
10 int fact(int n)
11 {
12     if(n==0)
13         return 1;
14     return n*fact(n-1);
15 }
16
17 int main()
18 {
19     int n;
20
21     printf("Enter the value of n\n");
22     scanf("%d",&n);
23     printf("The factorial of %d = %d \n",n,fact(n));
24     return 0;
25 }
```

input

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

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

## Program 1-

```
# 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++;
        }
        pop();
    } else {
        if (prcd(stack[top]) <= prcd(symbol)) {
            push(symbol);
        } else {

```

```

while (prcd(stack[top]) >= prcd(symbol)) {
    prefix[j] = pop();
    j++;
}
push(symbol);
}

}

}

```

```

while (stack[top] != '#') {
    prefix[j] = pop();
    j++;
}
prefix[j] = '\0';
}

```

```

void reverse(char array[30]) {

    int i, j;
    char temp[100];
    for (i = strlen(array) - 1, j = 0; i + 1 != 0; --i, ++j) {
        temp[j] = array[i];
    }
    temp[j] = '\0';
    strcpy(array, temp);
}

```

```
}
```

```
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) {  
    case '+':  
    case '-':  
    case '*':  
    case '/':  
    case '^':  
    case '$':  
    case '&':  
    case '(':  
    case ')':  
        return 1;  
        break;  
    default:  
        return 0;  
    }  
}
```

***Output-***



```
main.cpp
58
59     }
60 }
61
62 }
63
64 while (stack[top] != '#') {
65     prefix[j] = pop();
66     j++;
67 }
68 prefix[j] = '\0';
69 }
```

Input

```
main.cpp:24:1: warning: 'char* gets(char*)' is deprecated [-Wdeprecated-declarations]
/usr/include/stdio.h:638:14: note: declared here
main.cpp:24:11: warning: 'char* gets(char*)' is deprecated [-Wdeprecated-declarations]
/usr/include/stdio.h:638:14: note: declared here
main.cpp:(.text+0x2e): warning: the 'gets' function is dangerous and should not be used.
Enter infix operation: (a+(b-c)*d)
+a*-bcd

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

## Program 4-

```
#include <stdio.h>

int hcf(int n1, int n2);

int main() {
    int n1, n2;

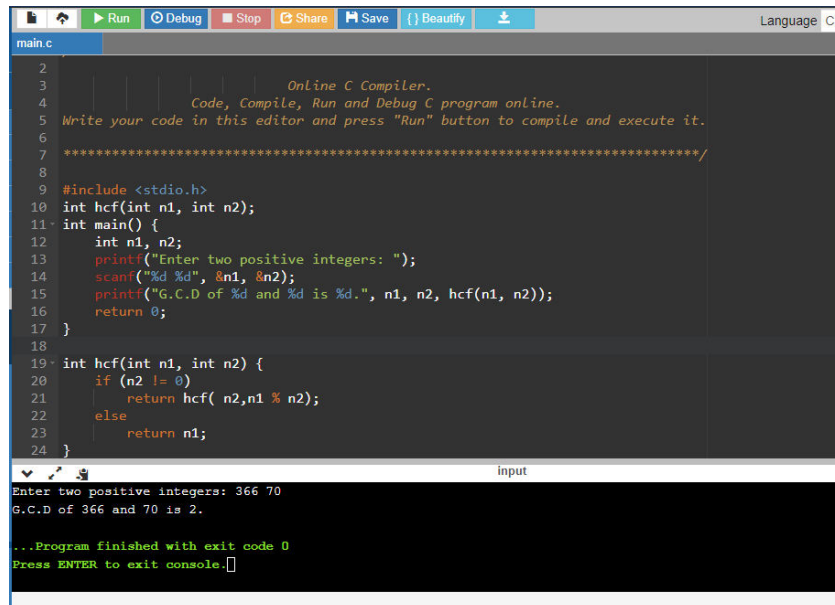
    printf("Enter two positive integers: ");
    scanf("%d %d", &n1, &n2);

    printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1, n2));

    return 0;
}

int hcf(int n1, int n2) {
    if (n2 != 0)
        return hcf( n2,n1 % n2);
    else
        return n1;
}
```

## Output-



The screenshot shows an online C compiler interface. At the top, there is a toolbar with buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The language is set to C. The code editor contains a C program for finding the G.C.D of two numbers. The output window shows the program's execution with input values 366 and 70, resulting in a G.C.D of 2.

```
main.c
2
3      Online C Compiler.
4      Code, Compile, Run and Debug C program online.
5      Write your code in this editor and press "Run" button to compile and execute it.
6
7      *****/
8
9      #include <stdio.h>
10     int hcf(int n1, int n2);
11     int main() {
12         int n1, n2;
13         printf("Enter two positive integers: ");
14         scanf("%d %d", &n1, &n2);
15         printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1, n2));
16         return 0;
17     }
18
19     int hcf(int n1, int n2) {
20         if (n2 != 0)
21             return hcf( n2,n1 % n2);
22         else
23             return n1;
24     }
```

input

Enter two positive integers: 366 70  
G.C.D of 366 and 70 is 2.  
...Program finished with exit code 0  
Press ENTER to exit console.