Applications of Stack

Stack is widely used in computer science. Following is the list of stack applications.

1. **Recursion**
2. **Evaluation of Expression**
3. **Stack Machine**

- **Recursion is one of the applications of stack.**
- **Function calling itself is called recursion and function is said to be recursive function.**
- **Also if function f1( ) calls function f2( ) and in turn function f2 ( ) calls function f1 ( ) then this function calls are known recursive function calls.**
- **There must exist base condition for which direct solution is available.**
- **When we call a function, it uses stack data structure as it pushes all the parameters in stack.**
- **Once execution of function is completed all the parameters are popped from stack.**
- **In recursive function calls a stack is created for each function and parameters are pushed and popped until function completes its execution.**

- **Following example demonstrates use of recursion**

$$\text{Fact}(N) = \begin{cases} = N * \text{Fact}(N-1) & \text{if } (N > 1) \\ = 1 & \text{if}(N = 0 \text{ or } N = 1) \end{cases}$$

For Example Fact(5) will create following stack.

F = Fact (5)

Fact (1) → F = 1
Fact (2) → F = 2 * Fact(1)
Fact (3) → F = 3 * Fact(2)
Fact (4) → F = 4 * Fact(3)
Fact (5) → F = 5 * Fact(4)

Stack of Recursive Function Calls

Algorithm Fact(N) : This function returns the factorial of given number N as N!. It is implemented as recursive function.

```
1. IF N <= 1
       Return (1)
2. F = N * Fact(N-1)
3. Return(F)
```

Write programs for following problems using recursion.

1. **Print 1 to N.**
2. **Print N to 1.**
3. **Print Nth Fibonacci Number.**
4. **Calculate X ^ Y using power(x, y) function**
5. **Calculate Sum(N) of first N natural numbers.**

Following C Program implement recursion to calculate factorial of a given integer number.

```
/* C Program to calculate factorial using recursion */
#include<stdio.h>
#include<conio.h>
int fact(int n);
void main()
{
        int n, f;
        clrscr();
        printf(“N = “);
        scanf(“%d”, &n);
        f = fact(n);

        printf(“%d! = %d\n”, n, f);
        getch();
}
```

```
int fact(int n)
{
        int f;
        if (n <= 1)
                return(1);
        f  = n * fact(n-1);
        return(f);
}
```

Sample Run:

```
N = 5
5! = 120
```

____