



Royal University of Bhutan



Unit IV – Part 03 (Recursive Function)

Lecture Slide



AS2022



Recursion

- **Recursion** is the is a process of calling a function by itself
- It is sometimes called '*circular definition*'-the process of defining something in terms of itself
- In recursion, the statements within the body of a called function in turn calls itself(the same function)



Recursion

- A recursive function must have an *if statement* somewhere to force the function to return without the recursive call being executed
- If there is no if statement to stop the recursion function, the function will never return



Recursion

- In every recursive solutions, it has two major required cases:

1. Base Case
2. Recursive Case

1. Base Case

- Acts as the terminating condition
- The case where the problem is simple enough to be solved directly
- No need of having any further calls to the same function



Royal University of Bhutan



Recursion

2. Recursive Case

- The problem at hand is divided into simpler sub-parts
- The function calls itself with sub-parts of the problem obtained in the earlier step
- The result is obtained by combining the solutions of simpler sub-parts



Recursion:Example 1:

```
main()
```

```
{
```

```
    printf("This is recursion  
example");
```

```
    main();
```

```
}
```



Royal University of Bhutan



Example 2: Recursive function to find the factorial

```
int fact(int n)
{
    int f=1;
    if(n==0)
        return f;
    else
        f=n*fact(n-
-1);
    return f;
}
```

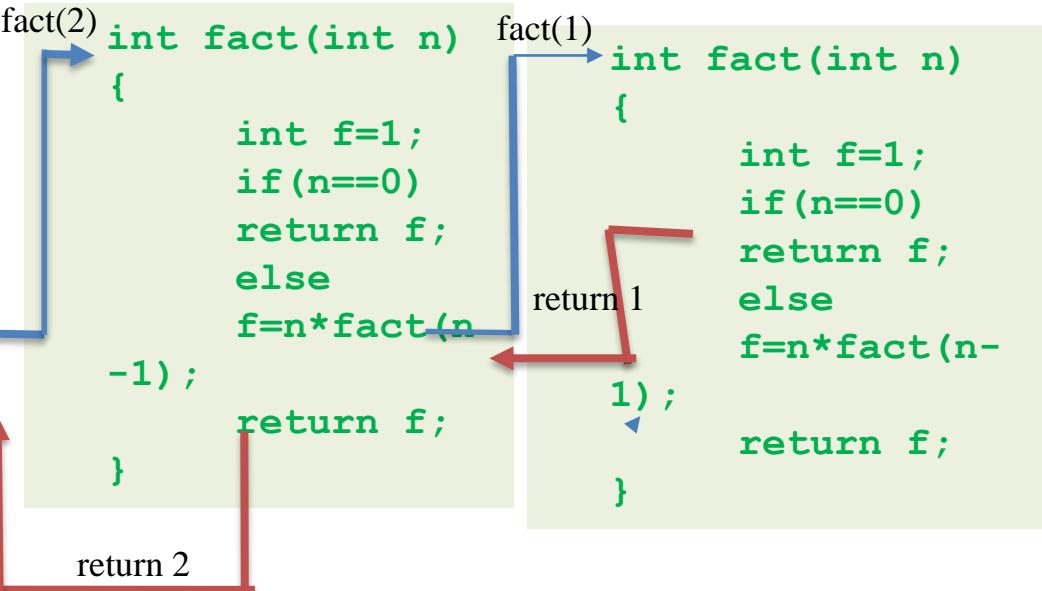
$$f=3*2*1$$

$$f=n*fact(2)$$

$$=3*2*fact(1)$$

$$=3*2*1$$

$$=6$$



Base Case: $n=1$, Recursive Case: factorial function that calls itself with smaller value of n



Recursion: Example 3

Recursive function to find the multiplication table.

```
void mult(int no,int min,int Max)
{
    if (min==Max+1)
        return;
    else
    {
        printf("\n%d*%d=%d",no,min,(no*min));
        mult(no,++min,Max);
    }
}
```



Recursion: Example 4

A recursive function repeatName() that calls itself till a length word/string is entered.

```
void repeatName()
{
    char n[50];
    printf("\nName:");
    scanf("%s",n);
    if(strlen(n)==1) //Base Case
        return;
    else
    {
        printf("\nName entered is %s",n);
        repeatName(); //Recursive call
    }
}
```



Home Assignment

1. Write a program to calculate x^y using recursion
2. Write a program to print the Fibonacci series using recursion



Thank you