

Lab Sheet 06

(1) What is a recursive method. Briefly explain.

A recursive method is a programming technique where a function calls itself to solve a problem by breaking it down into smaller, more manageable subproblems. Each recursive call operates on a reduced input and the process continues until a base case is reached, which is a simple case where the solution is known directly without further recursion.

(2) What is identified as an iteration. Briefly explain.

Repeated execution of a set of instructions is called iteration. It is the act of repeating a process to perform specific action on a collection of elements, one at a time, each repetition of the process is called iteration. The results of one iteration are used as the starting point for the next iteration until the condition is met. Iteration is most commonly expressed using loop statements.

- For Statement

- While Statement

- Do-While Statement.

(3) What is Factorial and fibonacci. Show how they can be used both as recursive.

The Factorial sequence calculates the product of all positive integers up to a given number.

Fibonacci sequence generates a series of numbers where each number is the sum of the two preceding ones.

Factorial - Recursive

```
#include <stdio.h>

int factorial recursive(int n){
    if (n==0)
        return 1;
    else
        return * Factorial recursive(n-1);
}
```

```
int main(){
    int num = 5;
    int factorial = Factorial recursive(num);
    printf("Factorial of %d is %d\n", num, factorial);
```

Factorial - Iteration

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

```
int factorial iterative(int n){
```

```
    int result = 1;
```

```
    for(int i=1; i<=n; i++){
        result *= i;
    }
```

```
    return result;
}
```

```
int main(){
```

```
    int num = 5;
}
```

RATHNA

Fibonacci - Iterative

```
#include <stdio.h>

int Fibonacci_iterative(int n){
    if(n<=1)
        return n;
    int prev_num = 0;
    int current_num = 1;
    for(int i=2; i<=n; i++) {
        int new_num = prev_num + current_num;
        prev_num = current_num;
        current_num = new_num;
    }
}
```

Fibonacci - Recursion

```
public class RecursiveFibonacci{
    public static void main(String[] args){
        for(int i=0; i<11; i++){
            System.out.print(Fibonacci(i) + " ");
        }
    }

    public static long Fibonacci(int n){
        if(n<=1){
            return n;
        }
        return Fibonacci(n-1) + Fibonacci(n-2);
    }
}
```