No 2023/07/05

Lab sheet (6) / Tute (8)

Iterations and Recursion.

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

A process in which a function calls itself, directly or indirectly. That corresponding function is called a recursive function. It helps to reduce the code length which makes easier approaches for coding. Ex:- factorial numbers. They use selection structures.

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

Iteration refers to repeating a particular steps until it meets the particular condition is met successfully. They're mostly used in linear programs where large numbers of variables are involved. Ex:- loops such as for, do-while and while.

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

- Factorial of a non-negative integer is the multiplication of all positive integers smaller than or equal to 6.

For ex:- factorial 6 is 6 * 5 * 4 * 3 * 2 * 1 which is 720.

- Fibonacchi is a data structure for priority queue operations, consisting of a collection of heap-ordered trees.

No:

Date: / /

- The ~~fibonacchi~~ factorial sequence calculate the product of all paitive integers up to given number.

- The fibonacchi sequence generates a series of numbers where each number is the sum of the two precending ones

```
# Pnculde <stdio.h>
Pnt factorial iterative{(int n)}.

{
    int result = 1 ;
    for (int j = 1 ; j < n ; j++)

    { result = j }.

    return result;
}

Pnt main ()
{
    Pnt num = 5 ;
    Int factorial = factorial recursive num);

    printf (" Factorial of "%d" is "%d\n" num.
    factorial);
    return 0;
}.
```

```
# include <stdio.h>
int fibonacci_recursive (int n)
{
    if (n <= 1)
    return n;

    else
        return fibonacci_recursive (n-1) +
                fibonacci_recursive (n-2);
    ;

    int main()
    {
        int num = 7;
        int fibonacci = fibonacci(num);

        return 0;
    }

# include <stdio.h>
int fibonacci_iterative (int n)
{
    if (n <= 1)
        return n;

    int pre_num = 0;
    int current_num = 1;
for (int i=2 ; i <= n ; i++)
{   int new_num = pre_num + current_num;
    pre_num = current_num;
    current_num = new_num;
}.
```