

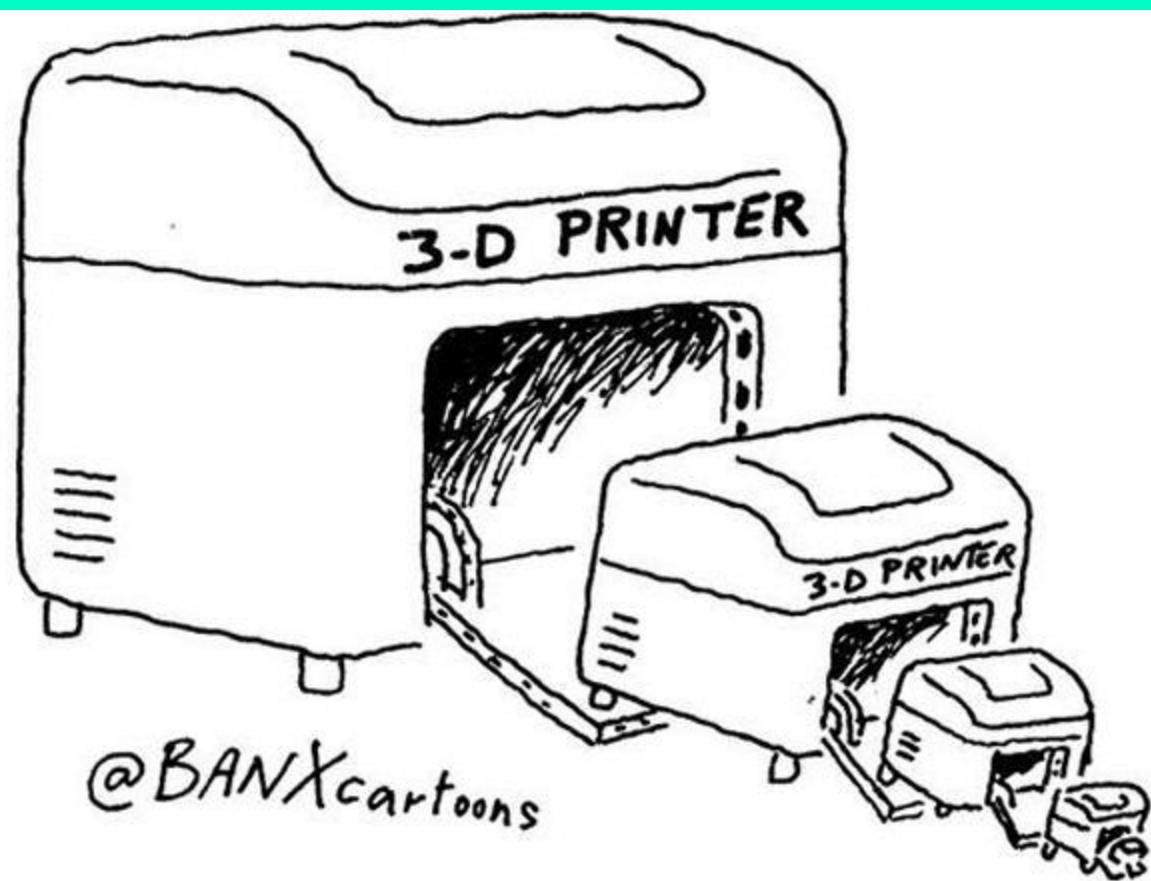
# ECE 220 REVIEW

**And we're back... .**

***To understand recursion,  
one must first understand  
recursion.***

*-STEPHEN HAWKING*

***My personal Favorite :***  
***To loop is human, to***  
***recurse is divine***



@BANXcartoons

# WHAT IS RECURSION THOUGH?

In programming terms, it basically means there is a function that just calls itself. That is it, no magic to it. The hard part is how to call that function.

In layman terms:

Read this sentence and do what it  
says twice.

## FUNDRAISING : ITERATION VS RECURSION

**Problem:** You have to collect \$1000 for charity. You can only ask for a penny from one person.

**Iteration:** Either you go to 100,000 people and ask for a penny each. (Ouch!)

**Recursion:** If you are asked to collect a penny, give a penny to the person who asked you for it.

Otherwise:-

- 1) Visit 10 people and ask them to each raise  $1/10$ th of the amount of money that you have been asked to raise.
- 2) Collect the money that they give you and combine it into one bag
- 3) Give it to the person who asked you to collect the money.

BASIC EXAMPLE:

THE FIBONACCI SERIES

$$\text{FIB}(N) = \text{FIB}(N-1) + \text{FIB}(N-2)$$

$$\text{FIB}(0) = 1$$

$$\text{FIB}(1) = 1$$



IMPORTANT:

ALWAYS REMEMBER TO HAVE A BASE CASE!!

ONLY 3 THINGS MATTER:

1) BASE CASE

2) WHAT YOU PROCESS

3) THE RECURSIVE CALL

LET'S DO SOME QUESTIONS

# EASY QUESTION-1

```
#define LIMIT 1000
void fun2(int n)
{
    if (n <= 0)
        return;
    if (n > LIMIT)
        return;
    printf("%d ", n);
    fun2(2*n);
    printf("%d ", n);
}
```

If we call `fun2(100)` ; it prints  
100,200,400,800,800,400,200,  
100

## EASY QUESTION-2

```
int fun(int a[],int n)
{
    int x;
    if(n == 1)
        return a[0];
    else
        x = fun(a, n-1);
    if(x > a[n-1])
        return x;
    else
        return a[n-1];
}

int main()
{
    int arr[] = {12, 10, 30, 50, 100};
    printf(" %d ", fun(arr, 5));
    getchar();
    return 0;
}
```

We get 100 as it finds the maximum number of the array.

## EASY QUESTION-3

- A) COMPUTE SUM OF NATURAL NUMBERS UPTO N.
- B) FIBONACCI
- C) TRIBONACCI

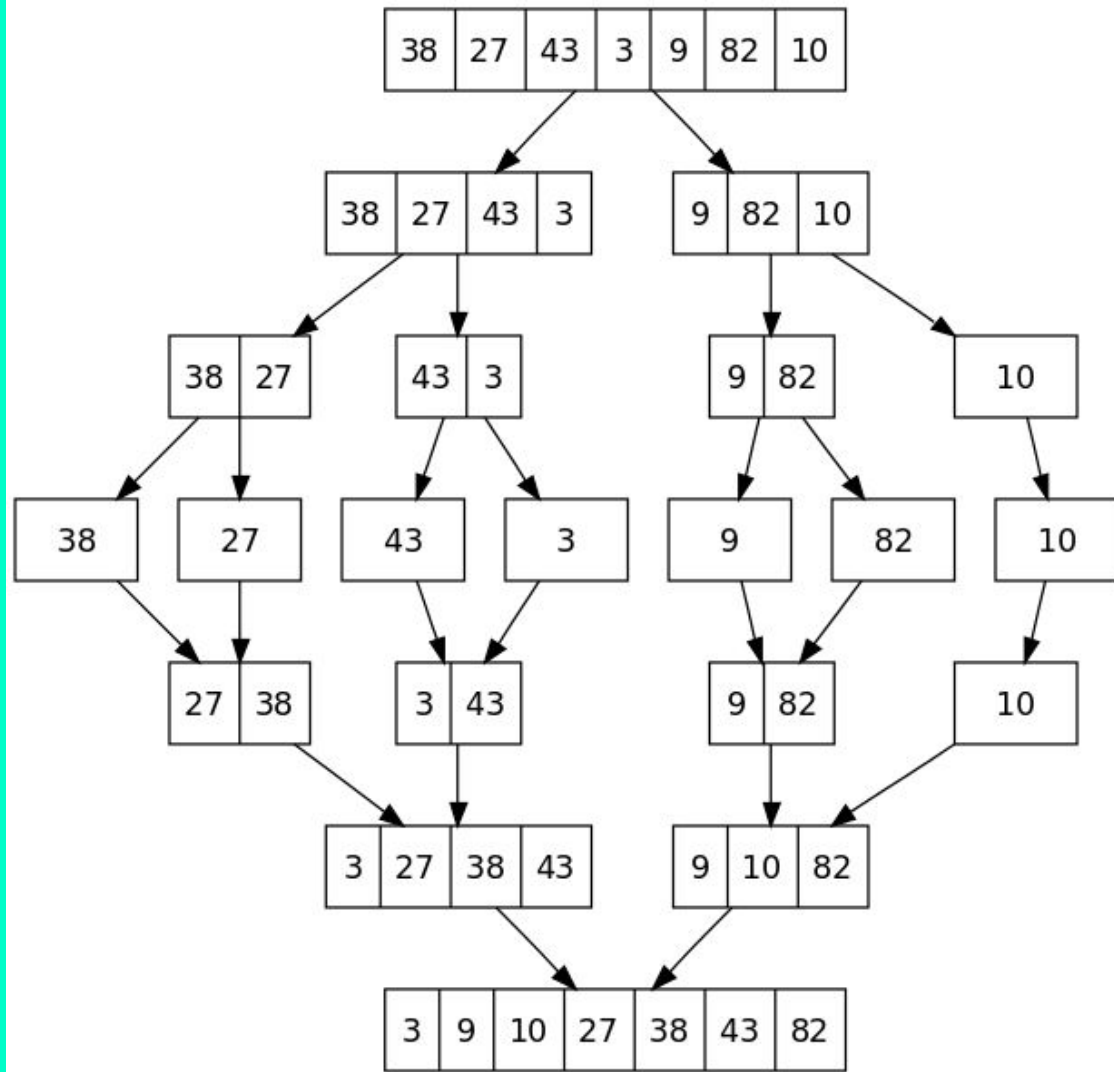
## MEDIUM QUESTION-1

PROBLEM 3 : RECURSION ON MT2\_PRACTICE\_MIDTERM\_1



## MEDIUM/HARD QUESTION-2

WRITE MERGE-SORT



```
MergeSort(arr[], l, r)
```

```
If r > l
```

```
1. Find the middle point to divide the array into two halves:
```

```
middle m = (l+r)/2
```

```
2. Call mergeSort for first half:
```

```
Call mergeSort(arr, l, m)
```

```
3. Call mergeSort for second half:
```

```
Call mergeSort(arr, m+1, r)
```

```
4. Merge the two halves sorted in step 2 and 3:
```

```
Call merge(arr, l, m, r)
```