Day: Date: / /

Recursion

A recursive function is one that calls itself.

e.g

```
1   void print () {
2
3       cout << This is infinite recursion aka Stack
4               overflow" << endl;
5
6       print ();
7   }
```

The above code will print the message infinite times because there is no condition to stop it. This condition is called a base case:

eg.

```
1   void print ( int n) {
2
3       if (n <= 0) {
4           return; }
5       cout << " This will stop \n";
6       print (n-1);
7   }
8
```

Yousuf

Day: Date:

for n=5, the message will be printed 5 times.

Properties of recursion:

- It must have a base case
- The recursive must change state and move towards base case.
- The recursive function must call itself

Using stack to show recursion.

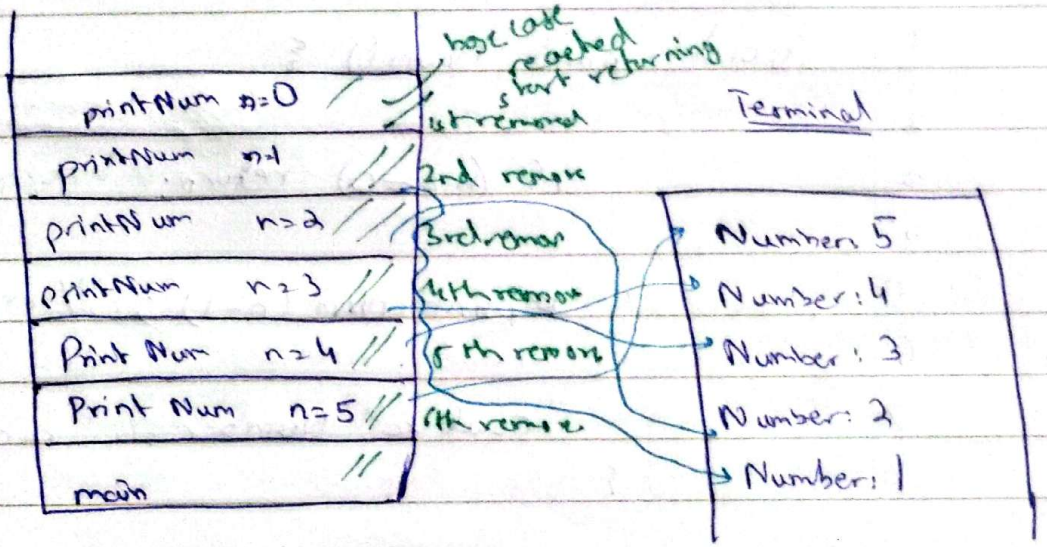
- It is always a good idea to make a stack and understand what is really happening.

1)

```
void printNum(int n) {
    if (n<=0) return; // base case
    cout << "Number: " << n << endl;
    printNum(n-1); // recursive case
}
```

Day: Date: / /

for n = 5



In this case we can see that whenever a case is called, the statement is printed with it. This is case of First in, first out.

- Whatever came in first printed it's content came out first.

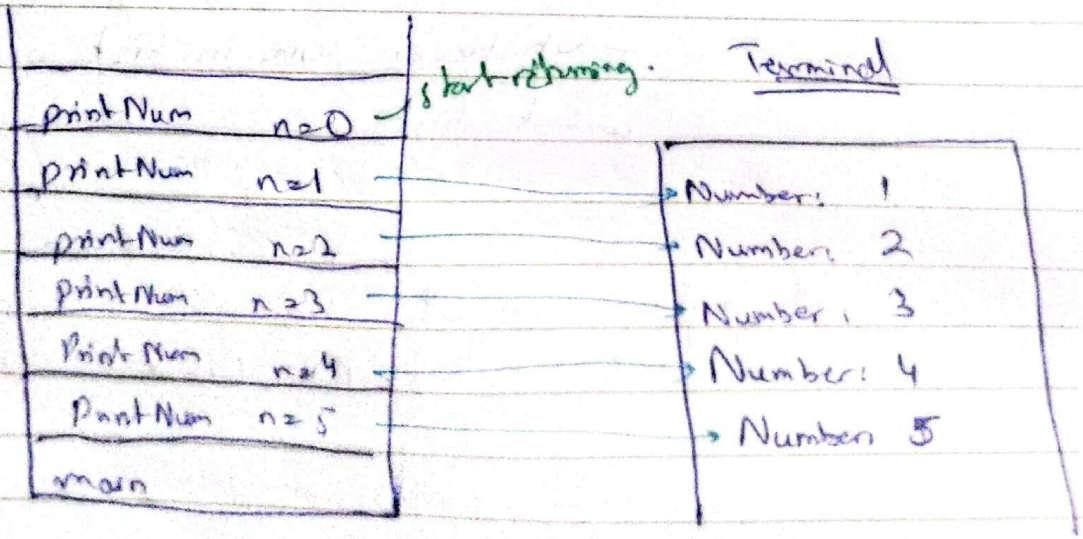
There can also be another case. First in, last out. SEE NEXT PAGE.

Yousaf

Day: Date: / /

```
1   void printNum() {
2
3       if (n <= 0) return; // base case
4
5       printNum(n-1); // recursive case.
6
7       cout << " Number: " << n << endl;
8   }
```

- Now in this program the statement comes after recursive case.



This is a example of First in, last out

Day: Date: / /

How? let's see a simpler version for n = 4, n=0.

```
void printNum(int n){
    if (n>=0) return;

    printNum(n-1); # This line ends as we have returned. now go to next line

    cout << " Number: " << n << endl;
}
```

- n=1 so it is printed.

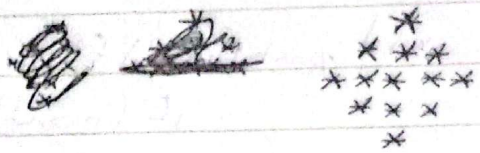
- When we go on n = 0, we return and go back to printNum(1) where n = 1.
- when n=0 returns, the printNum(1) has done its job and ends and we go to next line and the statement is printed.
- Then after this line Number: 1 is printed the function ends, and in stack we can see that we go back to n=2 or printNum(2). and this will now print this continues like this.

Yousaf

Day: Date: / /

1

2) Using both cases: Printing a diamond:

e.g for n = 3 : 

```
void spaces (int n, int max) { //helper func
    if (n == max) { // A case so that no extra spaces
                    at end are printed.
        return;
    }
    if (n <= 0) { return }; //True base cases

    cout << " ";
    spaces (n-1, max).
}

void triangle (int max, int n) { //helper function
    if (n >= max)
        { return; }

    cout << " * ";
    triangle (max, n+1);
}
```

Day: Date: / /

```
void printpattern(int n, int max){
    if (n<=0) return;

    spaces(n-1, max);
    triangle(max, n-1);
    triangle(max, n);

    cout<<endl;

printpattern(n-1, max);

    spaces(n, max);
    triangle(max, n);
    triangle(max, n+1);

    if(n!=max){
        cout<<endl;
    }
}
```

first in first out
Upper half

first in last out.
lower half.