Summarization

Topic: Recursion

Recursion is a process where the function calls itself again.

How do I write a recursive function?

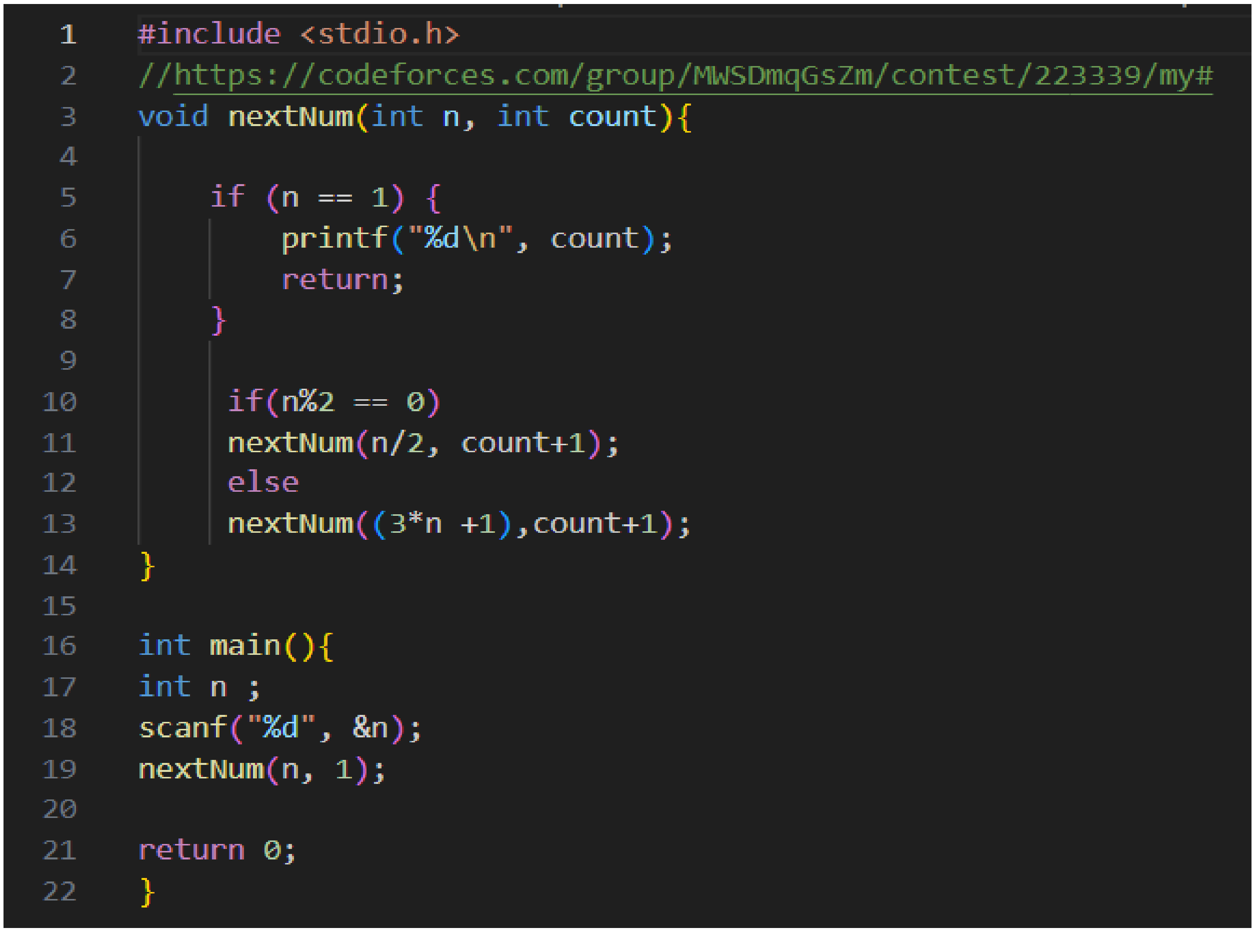
1- Divide the problem into smaller sub problems.
2- Determine the base condition where the recursion stops.

Types of recursion functions:

- Direct Recursion: The function calls itself again directly.
- Indirect Recursion: The function calls itself through another function.
- Tail recursion: The recursive call is the last operation in the function.
- Non-Tail recursion: The recursive call is followed by additional operations.

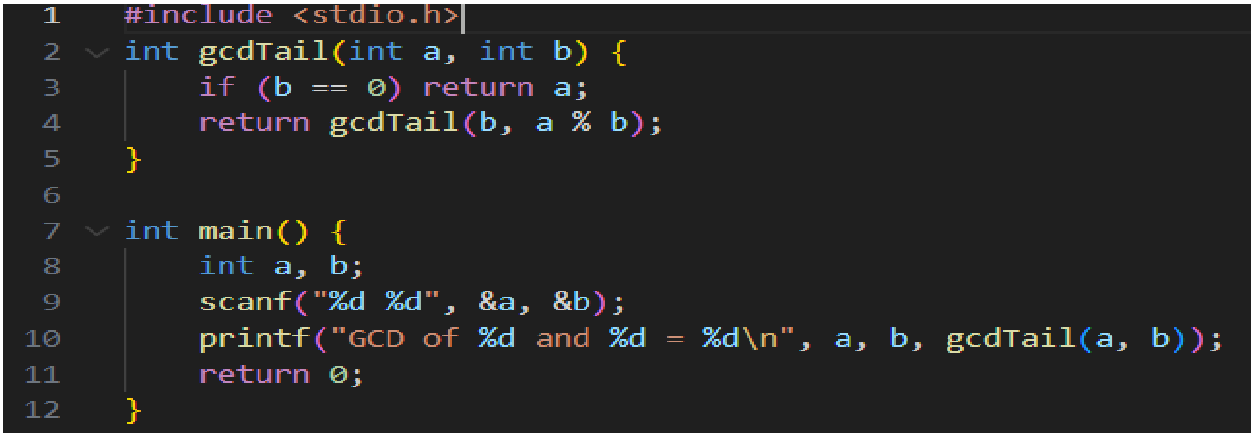
Examples:

In problem 8 the function’s type is direct Recursion & non-Tail Recursion (as the recursive call is not the last operation).



```c
#include <stdio.h>
//https://codeforces.com/group/MWSDmqGsZm/contest/223339/my#
void nextNum(int n, int count){

    if (n == 1) {
        printf("%d\n", count);
        return;
    }

     if(n%2 == 0)
     nextNum(n/2, count+1);
     else
     nextNum((3*n +1),count+1);
}

int main(){
int n ;
scanf("%d", &n);
nextNum(n, 1);

return 0;
}
```

A different approach to problem 7 is using the Euclidean Algorithm Where the type of recursion here will be direct and tail recursion.



```c
#include <stdio.h>
int gcdTail(int a, int b) {
    if (b == 0) return a;
    return gcdTail(b, a % b);
}

int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    printf("GCD of %d and %d = %d\n", a, b, gcdTail(a, b));
    return 0;
}
```