

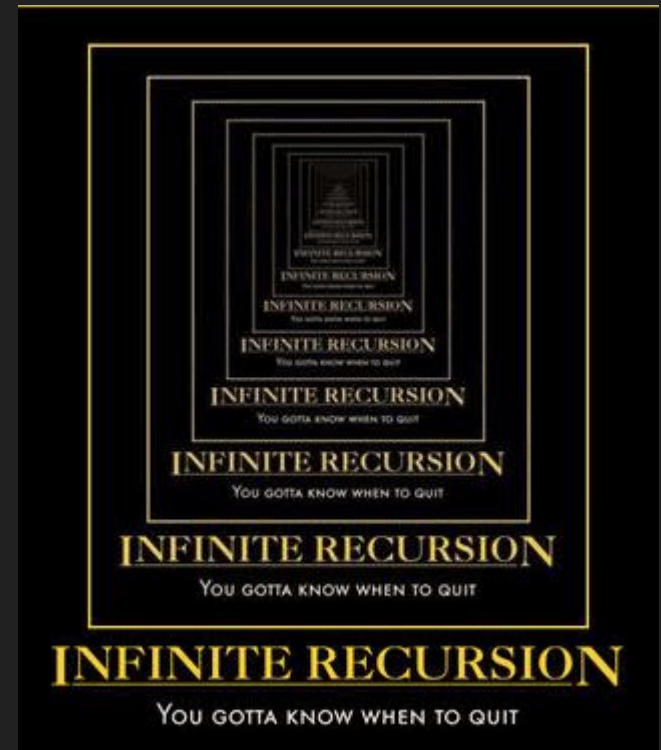
CENG 113 – Programming Basics

Lab 13

Recursions

Recursive Functions

- A function **calls itself** one or more times.
- Solves **smaller instances of the same problem** and combines the results.
- **Base Case (Exit Condition):** Where the problem can be solved without further recursion.
- Recursion can lead to an **infinite loop**, if the base case is never met.



Exercises

Harmonic Sum

- Write a recursive function that calculates the harmonic sum for the first **n** terms.

Example:

`get_harmonic_sum(5)` → $1 + 1/2 + 1/3 + 1/4 + 1/5$

Reversed List

- Write a recursive function to reverse a list.

Example:

```
get_reversed_list([1,2,3]) → [3,2,1]
```

Count Digits

- Write a recursive function `count_digits(n, k)` that counts the number of digits with value *k* for *n*.
- Write another recursive function `count_digits_upto(n, k)` that uses `count_digits` to count total number of digits with value *k* in $[0, n]$.

Example:

```
count_digits_upto(15052, 5)      →      2  
count_digits_upto(13, 1) → 6
```

Binary Search

- Write a recursive function `binary_search(arr, l, r, k)` that performs binary search over `arr`.

l : Index of the beginning element

r : Index of the ending element

k : Searching element

Example:

`binary_search([1,3,6,7,9], 0, 4, 6) → 2`

`binary_search([1,3,6,7,9], 0, 4, 5) → -1`

Egg Dropping (Bonus)

- Write a recursive function `egg_drop(n, k)` that returns the minimum number of trials required (in worst case) to find f .

n : Total number of floors in the building

k : The number of eggs to be dropped

f : The highest floor from which an egg can be dropped without breaking.

Details: <https://brilliant.org/wiki/egg-dropping/>