C++ Programming

Recursive Functions

Homework 2

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Homework 9: Right-Max

- Given array, change each element at position i to be the maximum of numbers from index i to end of array
- E.g. input 1 3 5 7 4 2 ⇒ [7, 7, 7, 7, 4, 2]
- Void left_max(int arr[], int len, int start_position = 0);

Homework 10: Suffix Sum

- Write a function that sums only the last N elements in an array.
- Define its signature
- Input [1, 3, 4, 6, 7], 3 ⇒ 17 (4+6+7)

Homework 11: Prefix Sum

- Write a function that sums only the first N elements in an array.
- Define its signature
- Input [1, 3, 4, 6, 7], 3 ⇒ 8 (1+3+4)

Homework 12: Is Palindrome

- Implement a function that decides if array is palindrome or not
- Define its signature

Homework 13: Is prefix

- bool is_prefix(string main, string prefix, int start_pos = 0)
- E.g. is_prefix(“abcdefgh”, “abcd”) ⇒ true
- E.g. is_prefix(“abcdefgh”, “”) ⇒ true
- E.g. is_prefix(“abcdefgh”, “abd”) ⇒ false

Homework 14: Trace

- Without running code on the right
  - Trace by hand: What does this method do?
  - What happens if we swapped lines 6 & 7?

```
3
4  void do_something(int n) {
5      if (n) {
6          cout << n % 10;
7          do_something(n / 10);
8      }
9  }
10
11 int main() {
12     do_something(123456);
13     return 0;
14 }
```

Homework 15: Count primes

- Int count_primes(int start, int end);
  - Compute how many primes between start & end, inclusive indices
- Don’t use loops at all
- Input
  - 10 20 ⇒ 4
  - 10 200 ⇒ 42
- Can u compute answer for [10, 5000000]?

Homework 16: Grid Sum

- Given a 2D array of numbers, all of them are positive distinct. Robot start from (0, 0). It can move to the right or left or diagonal. It will select one direction: the maximum. Print the total path sum of this robot
  - int path_sum(int grid[100][100], int row, int col, int ROWS, int COLS)
- Input
  - 3 3
  - 1 7 8
  - 2 10 11
  - 20 5 9
- Output: 31 (from 1 + 10 + 11 + 9)
  - Robot start at (0, 0). 3 possible values (2, 7, 10). Max 10, so go to this cell
  - Then 3 possible values (5, 9, 11). Go to 11. Then only 9 available

Homework 17: Fibonacci

- Implement fibonacci: Int fibonacci(int n)
  - Recall fibonacci sequence: 1 1 2 3 **5 8 13** 21 35
  - E.g. fibonacci(6) = 13
  - Recall that: fibonacci(n) = fibonacci(n-1) + fibonacci(n-2). E.g. fib(6) = fib(5)+fib(4) =13
    - So it calls 2 subproblems of its type
- Can u compute fibonacci(40)? fibonacci(50)? Why? Any work around? Hint: Array

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."