

# 1 Simple Algorithms

- Please submit a single file `FIRSTNAME_LASTNAME_simple.py` for this assignment.
- 
- Except for the first problem and the first part of the second problem, all of these problems require you to implement some existing algorithm.
  - For the sorting algorithms, I have included a link to a resource. Others you can google and understand easily enough.
  - Try to avoid looking at the code, and just understand the concept and then code it - for just the concept Wikipedia is pretty good. Otherwise don't worry, there are lots of problems in the world, so even if you look at the answer now - its not a big deal :)
- 

1. Write a function `check_prime(num)` to check if an integer is prime.
  2. Write a function `find_primes(high)` that prints all prime numbers less than high. Function should fail/exit for invalid values of high. Read about the Sieve of Eratosthenes.
  3. Write a function that takes a number `k` and a list of sorted numbers `L`, and returns `True` if `k` is in `L` or `False` otherwise. You **cannot** use the membership operator. You have to search using the Binary Search algorithm. Write two versions - first using recursion - `binary_search_recurse(mylist,value)` and then using loops `binary_search_loop(mylist,value)` . You can google the binary search algorithm.
  4. Write a function `selection_sort(mylist)` that takes a list of numbers and sorts it using the selection sort algorithm. You can read about the selection sort algorithm online - here is [one resource](https://www.hackerearth.com/practice/algorithms/sorting/selection-sort/tutorial/) (<https://www.hackerearth.com/practice/algorithms/sorting/selection-sort/tutorial/>). Its basic idea is to find the minimum and put it at the front.
  5. Write a function `bubble_sort(mylist)` that takes a list of numbers and sorts it using the bubble sort algorithm. You can read about the bubble sort algorithm online - here is [one resource](https://www.hackerearth.com/practice/algorithms/sorting/bubble-sort/tutorial/) (<https://www.hackerearth.com/practice/algorithms/sorting/bubble-sort/tutorial/>). Its basic idea is to keep pushing large elements to the back in each inner loop iteration.
- 

If you're feeling confident,

6. Write a function `merge_sort(mylist)` that takes a list of numbers and sorts it using the merge sort algorithm. Merge sort is a recursive algorithm. The crucial step is taking two recursively sorted subarrays and "merge" them together by "interleaving" them. Read more about it online - here is [one resource](https://www.hackerearth.com/practice/algorithms/sorting/merge-sort/tutorial/) (<https://www.hackerearth.com/practice/algorithms/sorting/merge-sort/tutorial/>).

