Assignment III

1. You are given an array of size n. For the elements in it, calculate their sum.
2. You are given an array of size n and a key k. Report the first and last index in the array at which k can be found.
3. Take n integers from the user and for each of them display whether they are prime or not. Define function bool isPrime(int n)
4. Prefix sum array is defined inductively as:-

   pre[i] = pre[i-1] + a[i] ; pre[0] = a[0]

   Given an array of size n, calculate the prefix sum array.
5. Given an array of size n, reverse it without using any auxiliary array i.e. you can't define another array.
6. Given an array of size n. Sort it. You can't use the builtin function sort() in your code. After you have done it, learn about the sort() function in C++ STL.
7. Define bool isPalindrome(string s); which takes a string as an argument and returns whether it is a palindrome or not. First do it without using any library function (i.e. using just loops). Learn about reverse() function and then solve the problem with its help.
8. Define int largestIndex(int arr[], int n) which takes as parameter an array and its size and returns the index of the element which has the largest value.
9. Read about Pass By Value and Pass By Reference in C++. Which way is used in passing arrays to function in C++?
10. Read about variable scope. Can a variable declared inside a function be accessed outside of the function? Can a variable declared inside a for loop be accessed outside of the loop? Can a variable declared outside of the for loop be accessed inside the for loop? What are global variables? When will you use them?
11. You are given two matrices of order n x m. Store and add them.
12. You are given two matrices of order n x m and p x q respectively. Store and multiply them or state that they can't be multiplied.
13. Find the sum of first n natural numbers using recursion.
14. You are in a n x m 2d world. From (i, j), you can either move to (i+1, j) or (i, j+1) i.e. Down or Right. Find the number of ways in which you can move from (1, 1) to (n, m). Use recursion
15. You are in a n x m 2d world. From (i, j), you can either move to (i+1, j) or (i, j+1) i.e. Down or Right. But some cells are bad. You are given a 2d array a[n][m] where a[x][y] = 1 means that cell (x, y) is bad. Find the number of ways in which you can move from (1, 1) to (n, m) without passing through any of the bad cells.

16. Solve https://www.hackerrank.com/contests/code-flix-1-0/challenges/to-be-or-not-to-be-a-triangle
17. Solve https://www.hackerrank.com/contests/code-flix-1-0/challenges/roll-it
18. Solve https://www.hackerrank.com/contests/code-flix-1-0/challenges/string-rotation-15-1
19. Solve https://www.hackerrank.com/contests/codeflix-2-0/challenges/jerry-vs-piyush
20. Solve https://www.hackerrank.com/contests/codeflix-2-0/challenges/dragon-run
21. Learn about substr() for smooth implementation. Solve https://www.hackerrank.com/contests/code-flix-3-0/challenges/emojis-
22. Solve https://www.hackerrank.com/contests/code-flix-3-0/challenges/very-easy-addition-1
23. Solve https://www.hackerrank.com/contests/code-flix-finalexx/challenges/maximum-difference-6-1
24. Solve https://www.hackerrank.com/contests/code-flix-finalexx/challenges/naughty-jerry

25. Solve https://projecteuler.net/problem=4