# CS218- Data Structures
# Programming Assignment No. 1
# Fall 2020

## Instruction

This is the first programming assignment for the course CS218- Data Structures in the offering Fall 2020. The assignment comprises of three problems. It is suggested that you should start working on the assignment at your earliest. This seems a good amount of intellectual work required to complete it. Each question should be solved on Hackerrank platform, invitation to CS218-Programming Assignment will soon be sent to all of you.

Important Note: The assignment is for individual and there should not be any case of cheating. You can have discussion about any problem and approach among yourself but do not share code and instruction for any problem. Try to write the code at hackerrank alone. The hackerrank count down will start from Saturday Morning 8AM and will expire on October 12, 2020 at 9:00PM

## Due Date: October 12, 2020 21:00PM

## Problem 1: Boggle Boggle

Boggle Boggle is a word game in which a character grid of 10x10 is presented on your screen and you need to form a word from the given characters in the grid. There is a list of words given to you, all you need to find the given words in the grid. Only the adjacent moves are allowed once you start from a letter (at any place) of a grid. This is to say that only horizontal, vertical and diagonal moves are allowed for matching character from the word. There is no requirement for order appearance of each character unlike word-puzzles. Consider a grid of a game below:

```
M V D V F G C V M N
P R O P E R V B S P
P M L T X I R H I R
R L T H T F L R E O
L M G G E R P O R P
X C U G F C M O P E
B K T J A W J L O R
X B D N A J B H Y I
T C J O O C K A X L
P R O E X S P Y C O
```

For example, in the above grid there are three instances of the word "P R O P E R"

```
M V D V F G C V M N
P R O P E R V B S P
P M L T X I R H I R
R L T H T F L R E O
L M G G E R P O R P
X C U G F C M O P E
B K T J A W J L O R
X B D N A J B H Y I
T C J O O C K A X L
P R O E X S P Y C O
```

This is an example of scoring bonus rewards. Where detecting all three instances will get a player 3000 score. 1000 per found instance.

**Input file format:** The first line of the input file for this problem contains size of the puzzle n and m two integers both single digit (maximum 9 and 9) for a 10 by 10 grid. The next line contains the first row of n possible row with m alphabets all upper case. After the puzzle the next line give number of words you need to search maximum 10- words per problem. One word per line. From the next line words are given. A sample input is given below:

**Output file format:** The output file contains word and its score. Remember that for every instance 1000 score is rewarded. if the word is not present in the puzzle just place a zero score on the corresponding line of that word in output file.

| Input File | Output File |
|---|---|
| 9 9<br>M V D V F G C V M N<br>P R O P E R V B S P<br>P M L T X I R H I R<br>R L T H T F L R E O<br>L M G G E R P O R P<br>X C U G F C M O P E<br>B K T J A W J L O R<br>X B D N A J B H Y I<br>T C J O O C K A X L<br>P R O E X S P Y C O<br>1<br>P R O P E R | P R O P E R 3000 |

## Problem 2: Array Palindrome

Given an array of n positive integers. Assume all the array elements are non-zero. You need to decide whether the content of arrays all together (from left to right) form a palindrome. By definition a palindrome is a word, number, phrase, or other sequence of characters which reads the same backward as forward, such as madam, racecar.  In the given problem let an array of 4 integer given below forms a palindrome

**Input file format:** The first line of the input file contains N as positive integer giving you size of the array.   The array values available from next line, single array value per line.

**Output file format:** The output file contains a YES if the given array is an array palindrome or NO otherwise.

| Input File | Output File |
|---|---|
| 4<br>134<br>21<br>12<br>431 | YES |

## Problem 3: Sum of subsets

In computer science, the subset sum problem is an important decision problem in complexity theory and cryptography. There are several equivalent formulations of the problem. The problem is NP-complete, meaning roughly that while it is easy to confirm whether a proposed solution is valid, it may inherently be prohibitively difficult to determine in the first place whether any solution exists. We intentionally introduced this problem to you as it is a good candidate for the idea of backtracking, we will keep the input size very small (N=10) to explore the entire solution space. The idea is quite simple. You are given an array A= {8, 6, 7, 5, 3, 10, 9} of positive integers, and a target value T. You need to check that is there a subset of elements in X that add up to T? Notice that there can be more than one such subset. Consider the given example A = {8, 6, 7, 5, 3, 10, 9} and T = 15, the answer is YES, because the subsets {8, 7} and {7, 5, 3} and {6, 9} and {5, 10} all sum to 15. On the other hand, if A = {11, 6, 5, 1, 7, 13, 12} and T = 15, the answer is NO.

**Input file format:** The first line of the input file contains N as positive integer giving you size of the array.   The array values available from next line, single array value per line.  The next line will be the target value T.

**Output file format:** The output file contains a YES if the given array a subset forms a sum T and NO Otherwise.

| Input File | Output File |
|---|---|
| 7 | YES. |
| 8 | |
| 6 | |
| 7 | |
| 5 | |
| 3 | |
| 10 | |
| 9 | |
| 15 | |