Assignment 08

Instructions

1. Write your functions according to the signature provided below in a python(.py) file and submit them to the autograder for evaluation.
2. The autograder has a limited number of attempts. Hence, test it thoroughly before submitting it for evaluation.
3. Save your python file as text(.txt) file and submit it to the canvas.
4. If the assignment has theoretical questions, upload those answers as a pdf file on the canvas.
5. Submission on the canvas is mandatory and the canvas submitted text file should be the same as that of the final submission of the autograder. If not, marks will be deducted.
6. No submission on canvas or no submission on autograder then marks will be deducted.
7. Access the auto grader at https://c200.luddy.indiana.edu.

Question 1

You are in charge of communications aboard a spaceship on a top-secret mission. For security reasons, mission control will send messages that have numeric values hidden within an array. You know that whenever you receive an array of numbers, the hidden number is the length of the largest subsequence such that all elements of the subsequence are exactly 7 away from at least one other element of the subsequence. Write code to automatically translate these arrays into integers.

Examples

```
Input: arr = [2,9,1]
Output: 2
```

Explanation: the subsequence <2,9> has 2 elements that are 7 apart from each other. Thus, the answer is 2.

```
Input: arr = [6,0,14,4,7,3]
Output: 3
```

Explanation: the subsequence <0,14,7> has 3 elements that are 7 apart from each other. Thus, the answer is 3.

Constraints:

- Solve using hashing
- There will be no duplicate values in the input array

Function

```
def translate_message(arr):
  #return int
  return value
```

Question 2

A blacksmith named Gary is crafting armor for the king. He received an order to make `n` shields of varying sizes. These orders came in an array of size `n` with each element of the list being a tuple of the form `[a`,b]‘ where a is the height of the shield and b is the length. However, before Gary could start crafting these orders, an evil wizard cast a spell on the list so that some of the elements were duplicated and then flipped within the array. Now Gary needs to find how many symmetric pairs are within the list so that he can remove the duplicates and make the correct shields for the king.

Write code that takes the list of orders and returns the size of the original list if all symmetric parts are removed.

Examples

```
Input: orders = [[1,2],[2,1]]
Output: 1
```

Explanation: The two elements in the array are symmetric pairs to each other. So one of them must be a duplicate. If we remove it, the list will have size of 1.

```
Input: orders = [[5,8], [3,2], [5,8]]
Output: 3
```

Explanation: There are no symmetric pairs in the list, so there are no duplicates. Thus, the size is still 3.

Constraints:

- Solve using hashing

Function

```
def find_order_size(orders):
  #return int
  return order_size
```

Question 3

On one fine day you decided to visit the circus nearby. In that circus, all the people are playing one game. In that game, you are displayed with a long string and you need to decide if that string has some repetition in it. You instantly notice that the string only has 4 unique characters in it W, X, Y, Z. Your job is to note down which strings of length 10 are repeating in the given string.

Examples

```
Input: circusString = "WWWWWXXXXXWWWWWXXXXXXWWWWWYYYZZZ"

Output: ["WWWWWXXXXX", "XXXXXWWWWW"]
```

Explanation: The given two sequences in the output array are present multiple times in the input string and have length = 10.

```
Input: circusString = "WWWWWWWWWWWWW"
```

```
Output: ["WWWWWWWWWW"]
```

Explanation: Only the given string with length 10 is repeating

Constraints:

- Solve using hashing
- Don't directly create hashMap with all substrings with length 10 as a key, instead use the Hashing function to detect the hash values for each substring of length 10

Function

```
def findCircusStrings(circusString):
  #return list
  return output
```

Question 4

You and your friends went to roam into the jungle. Suddenly you came across the tiger. You quickly started running to save your life. As you ran between the towering trees, you noticed peculiar markings on some of the trunks. Each tree had either a number or the ominous 'X' carved into its bark. It became clear that these trees held the key to your escape. Your friend shouted that the tiger has set some riddle. Tiger roars some random number at the start of the chase famously known as 'Tigers' Wish' and if you can find any 4 trees for which the sum of numbers carved on them is equal to the random number set by the tiger then the tiger lets you escape the jungle. The one thing to notice here is that, while adding up the values carved on the trunks of trees each tree has to be unique. If you couldn't find any of such trees then it's an unlucky day for you. You are given with the trees you are going to see through the chase and the tigers wish. Return 'Alive' if you escape the chase otherwise return 'Dead'.

Examples

```
Input: trees = ['2', 'X', '7', '-1', 'X', '5', '10', 'X', 'X'], tigersWish = 18
```

```
Output: 'Alive'
```

Explanation: The trees with markings 2, 7, -1, and 10 add up to tigersWish

```
Input: trees = ['X','X','X','1'], tigersWish = 8
```

```
Output: 'Dead'
```

Explanation: There is only a single tree in the jungle that has a number carved on it. Nothing can add up to 19.

Constraints:

- Solve using hashing
- Solve it in O(n^3)

Function

```
def aliveOrDead(trees, tigersWish):
  #return string
  return lifeStatus
```

Question 5

Now it is time for simulations as we end the course. Write the Python function which takes `n` as input and generates a point whose coordinates lie between `-1` and `1` for `n` number of times. You can use the random library of Python to generate the values. Out of these `n` points find the count of points whose value of $x^2 + y^2$ is less than 1 and assign it to the *count* variable.

1. Report the value of $4count/n$ for different values of n.
2. Using libraries of Python plot the n on the x-axis and $4 * count/n$ on the y-axis for different values of n
3. Write your observations for the above graph.

Note: Please submit the answer to this question in a PDF file.

The answer in the PDF file should contain the following,

1. Python code that is used to generate the table and graph.
2. A table containing n and $4count/n$ values
3. Graph for the above table.
4. Observations regarding the same.