**Code should be written in plain Javascript and production-level. Documentation, architecture etc will all be considered.**

**Part 1: Code logic, architecture and intelligence**  
  
**Task:**

Write a function, Function A, with the following characteristics:

* Takes in an array of points in string form [ex: *“([(1,2) (3,4)][(5,3) (7,8)])”]* where each element represents the coordinates of two points from which we can extract the corresponding line equations.
* Optional Parameter: Function A can also take in an array of random numbers that will be used as a key. If the array is not provided, the Function A will do the following.
* If array of random numbers is not provided, Function A generates an encryption key in the form of an an array of numbers [between numbers 1 and 30 (inclusive)] and uses the key to encrypt the array of points.
* After the key is used, the key is encrypted as well.
* The encrypted key and the input array are combined into one string and returned from function A.

Write another function, Function B, with the following characteristics:

* Receives an array that contains the concatenation of encrypted key & the encrypted point array.
* Separates the encrypted key and the encrypted point array.
* Then decrypts the key and then uses the key to decrypt the array.
* The function then uses the array of points to create the line equation for each pair of points.
* Finally, the function then creates an array where each entry is the string representation of the line equation and returns it.

**More detailed explanation of the task:** *Let’s use the following as a sample input for function A: “([(1,2) (3,4)][(5,3) (7,8)])”*

* Note: This data can be used to create 2 line equations.

*Now, let’s encrypt that input:*  
  
*Encrypting the input string:*

Generate an array with a length that is equal to the number of characters in the input, where each element is a random number between numbers 1 and 30 (inclusive). Iterate over the input string and:

* If corresponding random number is odd, subtract the random number from the current character (go left in accordance with ASCII table).
* If corresponding random number is even, add the random number from the current character (go right in accordance with ASCII table).

Example: For first 3 characters: if your input is “([(1,2) (3,4)][(5,3) (7,8)])” and your random number array looks like this [18,7,22,...] Then the first 3 characters of the input array will be transformed to “:” “T” and “>” accordingly.

**Encrypting the Encryption key (Random number array):**

*Once you are done encrypting your input string, now it is time to encrypt the encryption key itself. Here are the steps that you need to do to encrypt it.*

* For each number in the array:
  + Replace the digits with their corresponding alphabet character starting with 0 being equal to a and so on (Examples: 0 becomes a, 1 becomes b, 3 becomes d, 21 becomes cb, and so on.)
  + Convert each character to its string binary form and append space characters (“ “) at the beginning so that you end up with 8 character string for each of the characters.
* Then:
  + If the resulting string has an odd number of 1`s, replace all 0`s with the character “a” and all 1`s with character “b”, and after this, set the leading character (which was a space(“ “) character ) to 1.
  + If the resulting string has an even number of 1`s, replace all 0`s with character “b” and all 1`s with character “a”, and after this, set the leading character (which was a space(“ “) character ) to 0.
* Append the strings to each other if the original element had 2 characters.

**Example Walkthrough:**

*Let's say the first 2 numbers of your random number array were 1 and 14*

*Case 1: The number is 1*

* You replace 1 with it`s alphabetic value “b”.
* Then you convert this to its binary form, which becomes “1100010”.
* You append space characters until your string has 8 characters, so your string becomes (“ 1100010”).
* Since “ 1100010” has an odd number of 1`s, we replace all the 0`s with “a” and all the 1`s with “b” so the string becomes (“ bbaaaba”).
* Then, since the string “ bbaaaba” has odd number of 1`s, we replace the leading character with 1 so the final form of the string is (“1bbaaaba”).

*Case 2: The number is 14*

* You replace the 1 with it`s alphabetic value “b” and 4 with it`s alphabetic value “e” so your string becomes “be”.
* Then you convert each of these characters to their binary form so you end up with [“1100010”,”1100101”].
* You append space characters until your string have 8 characters so your strings become ( [“ 1100010”,” 1100101”].).
* The first of these has odd number of 1`s so it becomes “1bbaaaba”.
* The second one has even number of 1`s so it becomes “0aabbaba”.
* Finally, we append these to each other and get a final string of the form (“1bbaaaba0aabbaba”).

*So in order to encrypt the encryption key itself you need to repeat the procedure described above for each of the numbers in the encryption key array. The resulting array of strings is your encrypted encryption key.*

* Once the encryption is complete, the Function A concatenates the encrypted key array and encrypted input into one string by doing the following:
  + Join all the entries in the encrypted encryption key array by “,” and enclose the entire thing in square braces(“[“,”]”).
  + Enclose the encrypted string input into parentheses(“(“,”)”).

*For example: If your encrypted encryption key is an array containing 1aabbaba and 0aabbbba, and your encrypted string input is “34” the function A will return “[1aabbaba,0aabbba](34).*

* The output of Function A then can be passed to Function B. Function B decrypts the encryption key and then uses it to decrypt the string input.
* After this, Function B uses the data from the string input to generate line equations for each pair of points.

*Here is a solid example to better understand what are the inputs and expected outputs for each of the functions.*  
  
**Function A Example:**

* Suppose Function A receives the string “([(1,2) (3,4)][(5,3) (7,8)])”
* Suppose also that the random array generator comes up with this array:  
      [1,18,4,1,10,16,21,16,28,18,26,26,9,22,7,24,28,26,15,27,18,27,18,24,2,5,26,29]
* The function A encrypts the input string by using the random number array and produces the following encrypted input.

https://lh6.googleusercontent.com/m7l_HRbhdDmHVvEhiM8fGrzbYVRTkfBT6XfDmo0_QSVTU1t1GHyH84F1gR8KvPBDf0vCaRNzYYNTW1hWZ7JdpzHs8tTr5PA0NsBcBrsoXNh1IZe2nHWN0dYl2QdmT9OsYWW8InnH **(IMPORTANT: It might have some invisible characters here, but we do not care about that when we are working with code)**

* Function A then encrypts the key and produces the following array.

["1bbaaaba", "1bbaaaba0aababba", "0aabbaba", "1bbaaaba", "1bbaaaba1bbaaaab", "1bbaaaba1bbaabbb", "0aabbbaa1bbaaaba", "1bbaaaba1bbaabbb", "0aabbbaa0aababba", "1bbaaaba0aababba", "0aabbbaa1bbaabbb", "0aabbbaa1bbaabbb", "0aababab", "0aabbbaa0aabbbaa", "1bbabaaa", "0aabbbaa0aabbaba", "0aabbbaa0aababba", "0aabbbaa1bbaabbb", "1bbaaaba0aabbaab", "0aabbbaa1bbabaaa", "1bbaaaba0aababba", "0aabbbaa1bbabaaa", "1bbaaaba0aababba", "0aabbbaa0aabbaba", "0aabbbaa", "0aabbaab", "0aabbbaa1bbaabbb", "0aabbbaa0aababab"]

* At the end of this, Function A concatenates the encrypted encryption key and encrypted input into one string as explained [above](https://docs.google.com/document/d/1XSUh-8YPswqd-AVxv1eQrlhJTtyT5n3VUueT361ROCM/edit#bookmark=id.65h2duqid160) and returns it.

**Function B Example:**

* Following the previous example, the Function B will receive the concatenated string (encrypted encryption key + encrypted string input) and will separate the key from the input.
* Function B will then decrypt the key, will use the key to decrypt input, and will use the input to produce the following line equations.  
    
      For point pair **(1,2),(3,4)** The line equation will be **y=1x+1**   
      For point pair **(5,3),(7,8)** The line equation will be **y=2.5x-9.5**
* The function B will then put these equations into an array and return the answer.

What we’ll need from you:

* Please upload your code to a private repository on Gitlab, and add the following users as admins:
  + Menejyan.hovhannes
  + rodrigoborgesdeoliveira
  + Danielhaiem93
* Email [dan@classcalc.com](mailto:dan@classcalc.com) when you’re done, or if you have any questions. Include any of the following:
  + Github
  + Resume
  + LinkedIn
  + Cool things you’ve experienced
  + Whatever else you think is worth mentioning