At least parly, Is it possible to consider it as delivered on April 16, 2023, even though I was about 45 minutes late and actually delivered it on April 17, 2023?

Result:  
Test 1... inputs:

username: "sibel" - password1: "[rac()ecar]" - password2: 75

The username and passwords are valid. the Door is opening. Please wait...

Test 2... inputs:

username: "" - password1: "[rac()ecar]" - password2: 75

the username is invalid, it should be at least 1 character

Test 3... inputs:

username: "sibel1" - password1: "[rac()ecar]" - password2: 75

the username is invalid, it shouldcontains only letters

Test 4... inputs:

username: "sibel" - password1: "passs[]" - password2: 75

the password1 is invalid, it should be contains at least 8 character.

the password1 is invalid, it should be palindromable.

Test 5... inputs:

username: "sibel" - password1: "abcdabcd" - password2: 75

the password1 is invalid, it should have at least 2 brackets

Test 6... inputs:

username: "sibel" - password1: "[[[[]]]]" - password2: 75

the password1 is invalid, it should have at least 1 letter

Test 7... inputs:

username: "sibel" - password1: "[no](no)" - password2: 75

the password1 is invalid, it should be contains at least one letter of the given username.

Test 8... inputs:

username: "sibel" - password1: "[rac()ecar]]" - password2: 75

the password1 is invalid, it should be balanced

Test 9... inputs:

username: "sibel" - password1: "[rac()ecars]" - password2: 75

the password1 is invalid, it should be palindromable.

Test 10... inputs:

username: "sibel" - password1: "[rac()ecar]" - password2: 5

the password2 is invalid, it should be between 10 and 10000.

Test 11... inputs:

username: "sibel" - password1: "[rac()ecar]" - password2: 35

the password1 is invalid, it should be obtainable by summing demoniations.

Explain:

checkIfValidUsername method:  
Firstly it checks username validity and if lenght is greater than 1

Second it checks if first character is letter or not if it isn’t break terminate the recursion, if it is letter check finish condution that lenght of a substring is 1 (last letter of string) if not continue the recursion

containsUserNameSpirit method:  
Firstly create and initiliaze a stack for copy to password after that checks lenght of the password if it is less than 8 return false

Second Push all characters of password1 as lowercase to a stack, after the pushing checks if for each character of username (as lowercase) is in stack or not if one of them is in stack that’s meaning is we founded our character and it is valid for his case, if none of them is in stack return false and print the error

isBalancedPassword method:

Firstly crate and initiliaze a stack for checking, and create 2 flags to check is there any letter and bracket

Second there is a loop that iterates for each character of password, it checks brackets and characters. If the character is open bracket it is adding to stack if it is same but close versions of last added bracket loop pops last added bracket if it is close bracket but not same bracket the condition doesn’t provided and method returns false. And at the same time loop checks is there any letter and is there any none-letter-bracket character.  
  
At the end   
1- If there is no letter returns false  
2- If there is no brackets returns false  
3- If there is none-letter-bracket character returns false  
4- If the control stack is empty that meanind is all brackets are paired returns true

isPalindromePossible method:

Firstly it creates a StringBuilder for more effcient string building and deletes brackets from the string, after that for checking polindromablity it creates a boolean array (letter is used or not)  
  
Secondly it calls a recursivePolindrome function in that function:

Checks if index is last index (recursion completed succesfully), after that if curretn letter is paired before pass to next letter  
In the main part of recursion check it search for same character to check polindromablity (if there is another same letter it is polindrameable) if it’s founded continue and try other letters  
If There is no same letter and lenght of the password is odd for once (middle character) it allows to contiunue, otherwise returns false

isExactDivisionRecursive method:

Firstly it checks the interval of password if it is between 10 and 10000 it is ok but it is’nt between returns false. After that it calls isExactDivisionRecursive function:

Firstly if index = -1 this meaning is it is first call and if we do sum in first call we has to first number in result that’s why ignore first call  
If cuurent summation is equal the password that’s meaning is we achived our goal and obtained the password by summing demonitaions. If current sum greater than password we tried but we couldn’t achiave the password in this path.  
With loop and recursive call we check all posible summation for demonitions