Introduction to Programming

Labs – Week 11b

Exercise 1

Write a function `isValid()` to check whether a given string is a valid password, i.e., it obeys the following rules:

- A password must have at least ten characters.
- A password consists of only letters and digits.
- A password must contain at least two digits.
- A password must contain at least one letter.

A reference of useful `String` operations and other functions is provided at the end.

Exercise 2

Write a recursive function `removeX()` that given a string, compute recursively a new string where all the `'x'` characters have been removed. E.g.,

- `removeX("xaxb")` returns `"ab"`
- `removeX("abc")` returns `"abc"`
- `removeX("xx")` returns `""`

Hint: All occurrences of `'x'` in string `s` can be removed by removing `'x'` from first position (if exists) and recursively removing `'x'` from the rest

Exercise 3

Write a recursive function `evenDigits()` that accepts an integer parameter `n` and that returns the integer formed by removing the odd digits from `n`. The following table shows several calls and their expected return values:

Call	**Value returned**
`evenDigits(8342116)`	`8426`
`evenDigits(4109)`	`40`
`evenDigits(8)`	`8`
`evenDigits(-34512)`	`-42`
`evenDigits(-163505)`	`-60`
`evenDigits(7010496)`	`46`
`evenDigits(35179)`	`0`
`evenDigits(5307)`	`0`
`evenDigits(7)`	`0`

If a negative number with even digits other than `0` is passed to the method, the result should also be negative, as shown above when `-34512` is passed. Leading zeros in the result should be ignored and if there are no even digits other than `0` in the number, the method should return `0`, as shown in the last three outputs.

Exercise 4

Two strings are called anagrams if they contain same set of characters but in different order. E.g., “Debit card” = “Bad credit”, “Graduation” = “Out in a drag!”, “Election Results” = “Lies, Let's Recount!”, or “Software” = “Swear Oft”. Write a function `isAnagram()` which checks whether two given strings are anagrams of each other.

Exercise 5

Given a matrix of positive numbers (unsorted) m, an integer sum and another matrix p that filled with 0 all over, write a recursive function to check if there is a path inside m that the sum of it will be equal to sum.

The rules

- you can only travel to down, up , left or right in the array.
- after you've found the path, the matrix p will be filled with 1's on the correct path.
- there is only 1 path
- all other cells on p should be 0 after the method has finished.
- if there is no path to this sum you will leave p as you got him.

Example

```
int[][] p = {{0,0,0,0},
             {0,0,0,0},
             {0,0,0,0},
             {0,0,0,0}};
```

in the beginning. The matrix is:

```
int [][] hill = {{3,8,7,1},
                 {5,15,2,4},
                 {12,14,-13,22},
                 {13,16,17,52}};
```

If you call the method on sum = 23 the method will return true, and p will be:

```
int[][] p = {{1,0,0,0},
             {1,1,0,0},
             {0,0,0,0},
             {0,0,0,0}};
```

The function must be recursive.

Reference – String Operations & Character functions

- `s.length()` returns length of a string `s`
- `s.charAt(i)` returns the character at index `i` of string `s`
- `s.toLowerCase()` returns a copy of `s` with all characters converted to lowercase
- To check if two strings `s` and `t` are equal, write `s.equals(t)`
- `p.substring(i,j)` will return substring of `p` starting with the `i`th character and ending with `(j-1)`st character of `p`. E.g., `p.substring(1,p.length()-1)` will return copy of `p` with first and last character removed
- `p.substring(i)` return suffix of `p` starting from position of `i`. E.g., `p.substring(1)` will return `p` after removing first character of `p`
- You can check if `char c` is a whitespace (i.e., space, tab, or newline) with `Character.isWhitespace(c)`
- You can check if `char c` is a letter with `Character.isLetter(c)`
- You can check if `char c` is a digit with `Character.isDigit(c)`
- You can check if `char c` is a letter or digit with `Character.isLetterOrDigit(c)`
- A `char c` can be converted to `String` either by using `Character.toString(c)` or concatenating it with empty string `c+""`