Lab 14

Task 1 Write a recursive function that takes a positive integer n as input and returns the value of $1^2 + 2^2 + 3^2 + ... + n^2$.

Task 2 Write a function, "subsequences" that returns all the subsequences in a list of a string (which the function takes as an argument). For example, subsequences of "abc" might return ["abc","ab","bc","ac","a","b","c",""]. Note the empty string, which is also a valid subsequence.

This problem lends itself to an elegant recursive decomposition. Take the first letter of the word. We can form one set of subsequences that include that letter, and another set of subsequences that exclude that letter, and those two sets completely cover the set of possible subsequences.

Sample run:

```
Enter string: abc
Subsequences of abc = ["abc","ab","bc","ac","a","b","c",""].
```

Task 3 Write a function that gives change, i.e. it takes an amount as argument, and it prints out P's, N's, D's, and Q's - one for each penny (1 cent), nickle (5 cent), dime (10 cent) and quarter (25 cent) it gives out.

Ex: If Amount is 33, it would print out:

QNPPP

Your function should be recursive, and here's some simple rules to help you (let A be the amount you need change for):

1) If A is 0, don't give anything!
2) If A > 0, give out one of the largest denomination not exceeding A, then make
change for the remaining amount.

Task 4 In the previous task, the output was ordered from the highest value coin to lowest value coin. Change the above function to display from lowest value coin to highest value coin.

Ex: If Amount is 33, it would print out:

PPPNQ