Recursion

For this lab, you will write two recursive functions. The test program and header file are provided for you.

isPalendrome

Implement a function isPalindrome which takes a string as an argument and returns true if the word reads the same forward and backwards and false otherwise. For example, “racecar” and “noon” are palindromes, but “banana” is not. Your function should be recursive (i.e. it should call itself and have a base case). You will find the substr function of the string class useful in your solution.

printAllSubsets

Implement a function printAllSubsets which takes a string and prints all subsets of the string. For example, if the string was “dog”, the subsets would be:

dog do dg og d o g

Note that the order in which the subsets are printed does not matter, nor does the ordering of the letters within each subset (i.e. including “og” or “go” is fine, just not both). In order to implement this, printAllSubsets will be a wrapper function for your actual recursion. That is the prototype for printAllSubsets will be:

```
void printAllSubsets(string word);
```

For your recursive function you’ll need two strings, an input string holding the letters you’ve not yet considered adding to your output, and an output string holding the letters you have added to your outputs. Thus, the prototype for your recursive function should look like this:

```
void recPrintAllSubsets(string input, string output);
```

Your printAllSubsets function should just call recPrintAllSubsets with the initial word as the input and the empty string as the output

Testing your program

There is a test program provided in Moodle along with a recurse.h and a starting recurse.cpp. You are to write your functions in recurse.cpp and test them with the test program.