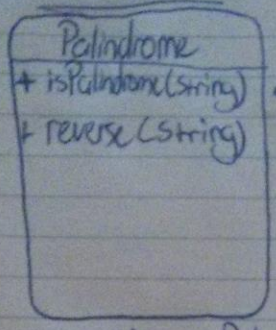
Prac 4 Design 05-04-14

Palindrome



How do these functions work -

bool isPalindrome(string): as in Prac 1 simply check each letter (up to length /2) against its corresponding letter from the end of the string.

i.e. racecar 1. r=r 2. a=a 3. c=c (don't need to check middle value always equals itself)

String reverse(string):

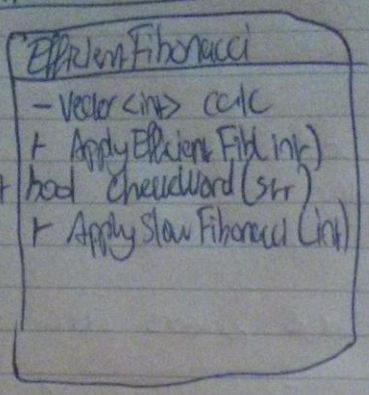
If the string is length 1 simply return it
If not take the string from index 1 to the end and call reverse on it (but add the letter at index 0 to this call)

i.e. reverse(temp) + string[0]
where temp = string 1 → length

eg reverse(man)

call reverse("an") + "m"
call reverse("n") + "a" + "m"
n is now length 1 so → return "n" + "a" + "m"

EfficientFibonacci



CheckWord :- iterate through each "letter" in the "word" and check if they are digits

SlowFibonacci: base cases int=0 return 0
int=1 or 2 return 1
else return slowFib(int-1) + slowFib(int-2)

EfficientFib:
Similar to above but this time check if the value at calc[num] not 0, if it isn't then return that value, if it is do as above but store the return value in check[nums]
This way instead of repeating calls it uses stored values (Bonus? The more this algorithm is run the faster it gets !!!)

TESTING:

- Test using a very massive INT (> 100) one would assume this breaches the limit of int
- Test using 0 as input to Fibonacci
- Test using negative input to Fibonacci
- Test with no input to Fibonacci
- Test