Problem 1

d ={'data':1, 'structures':1000, 'and':15}

update(d, 'data', 1)

print(d) #expect:{'data':2, 'structures':1000, 'and':15}

update(d, 'and', 5)

print(d) #expect:{'data':2, 'structures':1000, 'and':20}

update(d, 'algorithms', 1)

print(d) #expect:{'data':2, 'structures':1000, 'and':20, 'algorithms':1}

update(d, 'algorithms', 4)

print(d) #expect:{'data':2, 'structures':1000, 'and':20, 'algorithms':5}

In [9]:
```
def update(d, word, amount):
    if word in d:
        d[word] = d[word] + amount
    else:
        d.update({word: amount})
    return d

d = {'data': 1, 'structures': 1000, 'and': 15}
print(d)
print(update(d, 'data', 1))
print(update(d, 'and', 5))
print(update(d, 'algorithms', 1))
print(update(d, 'algorithms', 4))
```

```
{'data': 1, 'structures': 1000, 'and': 15}
{'data': 2, 'structures': 1000, 'and': 15}
{'data': 2, 'structures': 1000, 'and': 20}
{'data': 2, 'structures': 1000, 'and': 20, 'algorithms': 1}
{'data': 2, 'structures': 1000, 'and': 20, 'algorithms': 5}
```

Problem 2: A

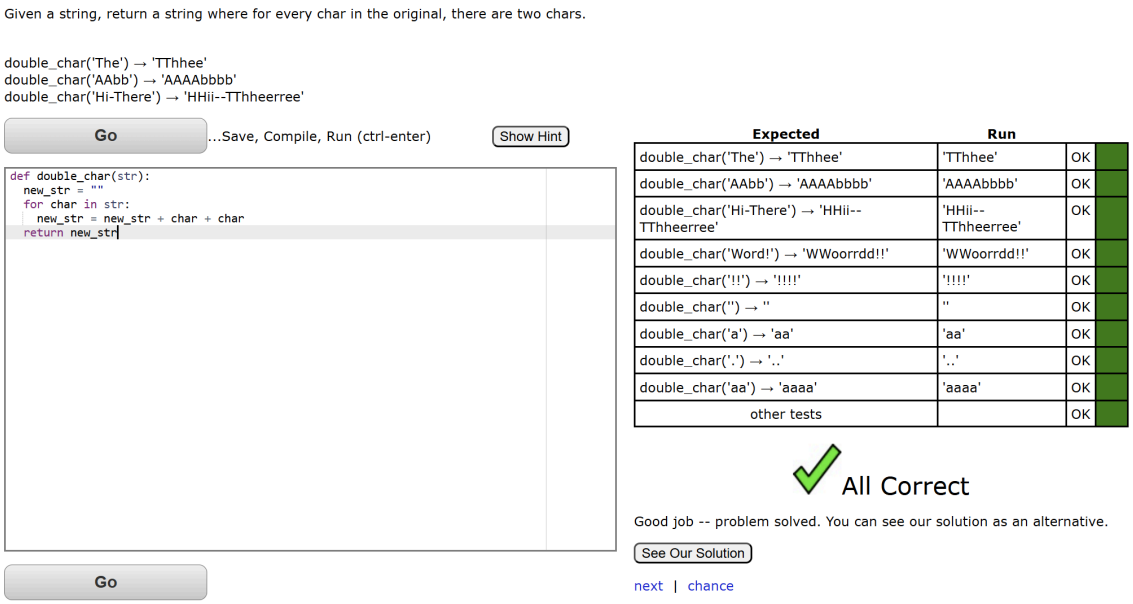
https://codingbat.com/prob/p170842

Purpose: Returns a given string where for every char in the original, there are two chars.

Input: str, string input

Output: the original string with every char doubled

Assumptions: None



Problem 2: B

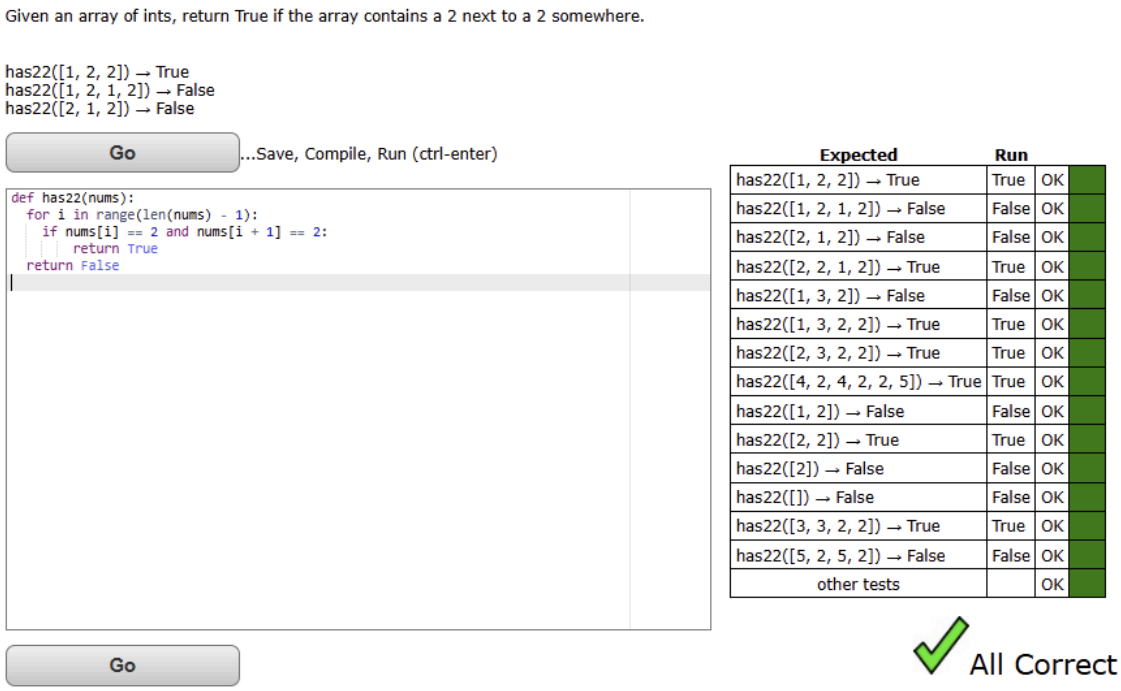
https://codingbat.com/prob/p119308

Purpose: Determines if an array contains a 2 next to another 2

Input: nums, list of numbers

Output: returns true if the array contains a 2 next to another 2, otherwise false

Assumptions: None



Problem 3

Choose one of the coding bat problems above and rewrite the solution using recursion. Submit a screen shot showing your recursive code and the results of running your code

In [8]:
```
#Purpose: Returns a given string where for every char in the original, there
#Input: str, string input
#Output: the original string with every char doubled
#Assumptions: None

def double_char(str):
  if len(str) == 1:
    return str + str
  else:
    return str[0] + str[0] + double_char(str[1:])

print(double_char('The'))
print(double_char('AaBb'))
print(double_char('Hi-There'))
```

```
TThhee
AAaaBBbb
HHii--TThheerree
```