

## P35

1.

Logic-1 > cigar\_party

prev | next | chance

When squirrels get together for a party, they like to have cigars. A squirrel party is successful when the number of cigars is between 40 and 60, inclusive. Unless it is the weekend, in which case there is no upper bound on the number of cigars. Return True if the party with the given values is successful, or False otherwise.

cigar\_party(30, False) → False  
cigar\_party(50, False) → True  
cigar\_party(70, True) → True

Go ...Save, Compile, Run (ctrl-enter) Show Hint

```
def cigar_party(cigars, is_weekend):  
    if is_weekend == False and cigars >= 40 and cigars <= 60:  
        return True  
    elif is_weekend and cigars >= 40:  
        return True  
    else:  
        return False
```

Expected	Run
cigar_party(30, False) → False	False OK
cigar_party(50, False) → True	True OK
cigar_party(70, True) → True	True OK
cigar_party(30, True) → False	False OK
cigar_party(50, True) → True	True OK
cigar_party(60, False) → True	True OK
cigar_party(61, False) → False	False OK
cigar_party(40, False) → True	True OK
cigar_party(39, False) → False	False OK
cigar_party(40, True) → True	True OK
cigar_party(39, True) → False	False OK
other tests	OK

✓ All Correct

Good job -- problem solved. You can see our solution as an alternative.

See Our Solution

2.

List-2 > count\_evens

prev | next | chance

Return the number of even ints in the given array. Note: the % "mod" operator computes the remainder, e.g. 5 % 2 is 1.

count\_evens([2, 1, 2, 3, 4]) → 3  
count\_evens([2, 2, 0]) → 3  
count\_evens([1, 3, 5]) → 0

Go ...Save, Compile, Run (ctrl-enter) Show Hint

```
def count_evens(nums):  
    cnt = 0  
    for num in nums:  
        if num % 2 == 0:  
            cnt += 1  
    return cnt
```

Expected	Run
count_evens([2, 1, 2, 3, 4]) → 3	3 OK
count_evens([2, 2, 0]) → 3	3 OK
count_evens([1, 3, 5]) → 0	0 OK
count_evens([]) → 0	0 OK
count_evens([11, 9, 0, 1]) → 1	1 OK
count_evens([2, 11, 9, 0]) → 2	2 OK
count_evens([2]) → 1	1 OK
count_evens([2, 5, 12]) → 2	2 OK
other tests	OK

✓ All Correct

Good job -- problem solved. You can see our solution as an alternative.

See Our Solution

next | chance

Python > List-2

3.

https://codingbat.com/prob/p119308

List-2 > has22

prev | next | chance

Given an array of ints, return True if the array contains a 2 next to a 2 somewhere.

has22([1, 2, 2]) → True  
has22([1, 2, 1, 2]) → False  
has22([2, 1, 2]) → False

Go ...Save, Compile, Run (ctrl-enter)

```
def has22(nums):
    is_found = False
    is_sequence = False
    for n in nums:
        if is_found and n == 2:
            is_sequence = True
        elif n == 2:
            is_found = True
        else:
            is_found = False
    return is_sequence
```

Go

Expected	Run
has22([1, 2, 2]) → True	True OK
has22([1, 2, 1, 2]) → False	False OK
has22([2, 1, 2]) → False	False OK
has22([2, 2, 1, 2]) → True	True OK
has22([1, 3, 2]) → False	False OK
has22([1, 3, 2, 2]) → True	True OK
has22([2, 3, 2, 2]) → True	True OK
has22([4, 2, 4, 2, 2, 5]) → True	True OK
has22([1, 2]) → False	False OK
has22([2, 2]) → True	True OK
has22([2]) → False	False OK
has22([]) → False	False OK
has22([3, 3, 2, 2]) → True	True OK
has22([5, 2, 5, 2]) → False	False OK
other tests	OK

✓ All Correct

4.

https://codingbat.com/prob/p108886

List-2 > sum67

prev | next | chance

Return the sum of the numbers in the array, except ignore sections of numbers starting with a 6 and extending to the next 7 (every 6 will be followed by at least one 7). Return 0 for no numbers.

sum67([1, 2, 2]) → 5  
sum67([1, 2, 2, 6, 99, 99, 7]) → 5  
sum67([1, 1, 6, 7, 2]) → 4

Go ...Save, Compile, Run (ctrl-enter)

```
def sum67(nums):
    sum = 0
    is_skip = False
    for num in nums:
        if num == 6:
            is_skip = True
        elif is_skip == True and num == 7:
            is_skip = False
        elif is_skip == False:
            sum = sum + num
    return sum
```

Go

Expected	Run
sum67([1, 2, 2]) → 5	5 OK
sum67([1, 2, 2, 6, 99, 99, 7]) → 5	5 OK
sum67([1, 1, 6, 7, 2]) → 4	4 OK
sum67([1, 6, 2, 2, 7, 1, 6, 99, 99, 7]) → 2	2 OK
sum67([1, 6, 2, 6, 2, 7, 1, 6, 99, 99, 7]) → 2	2 OK
sum67([2, 7, 6, 2, 6, 7, 2, 7]) → 18	18 OK
sum67([2, 7, 6, 2, 6, 2, 7]) → 9	9 OK
sum67([1, 6, 7, 7]) → 8	8 OK
sum67([6, 7, 1, 6, 7, 7]) → 8	8 OK
sum67([6, 8, 1, 6, 7]) → 0	0 OK
sum67([]) → 0	0 OK
sum67([6, 7, 11]) → 11	11 OK
sum67([11, 6, 7, 11]) → 22	22 OK
sum67([2, 2, 6, 7, 7]) → 11	11 OK
other tests	OK

✓★ All Correct