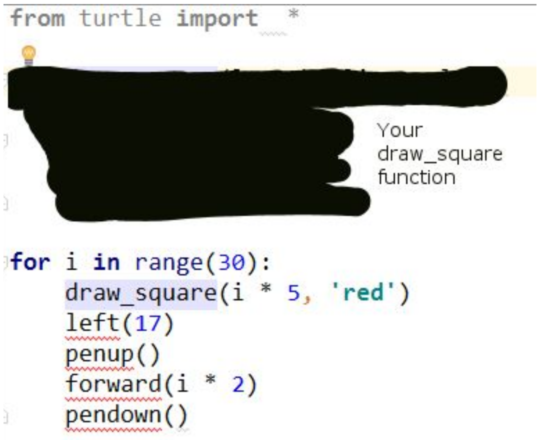
Turtle exercise

1. Write a Python function that draws a square, named `draw_square`, takes 2 input parameters: `length` and `color`, where `length` is the `length` of its side and `color` is the `color` of its bound (line color)
2. Now, another programmer named Huy be will use your code in exercise 1. He writes as follows:

```
for i in range(30):
    draw_square(i * 5, 'red')
    left(17)
    penup()
    forward(i * 2)
    pendown()
```

Copy this code into your editor, run the whole program and see what it draws:
Note: If your code does not run, try not to modify Huy be's code, modify your function instead



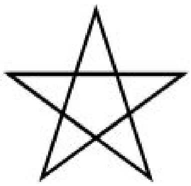
```
from turtle import *

[redacted]

for i in range(30):
    draw_square(i * 5, 'red')
    left(17)
    penup()
    forward(i * 2)
    pendown()
```

Your draw_square function

3. Write a Python function that draws a star, named `draw_star`, take 3 parameters: `x`, `y`, and `length`. Where `x`, `y` are the location of the star, `length` is the length of its side

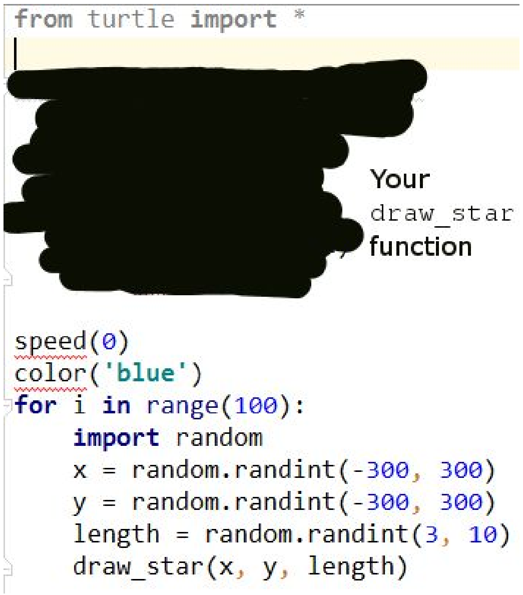


Hint: Turn 144 degree at each point

4. Again, your function will be used by other programmers like Hiep want to use your function, they writes as follows:

```
speed(0)
color('blue')
for i in range(100):
    import random
    x = random.randint(-300, 300)
    y = random.randint(-300, 300)
    length = random.randint(3, 10)
    draw_star(x, y, length)
```

Copy this code into your editor, run the whole program and see what it draws:



```
from turtle import *

[redacted]

speed(0)
color('blue')
for i in range(100):
    import random
    x = random.randint(-300, 300)
    y = random.randint(-300, 300)
    length = random.randint(3, 10)
    draw_star(x, y, length)
```

Your `draw_star` function

Explain the `random.randint(…)` statement, what it is, and how to use it?

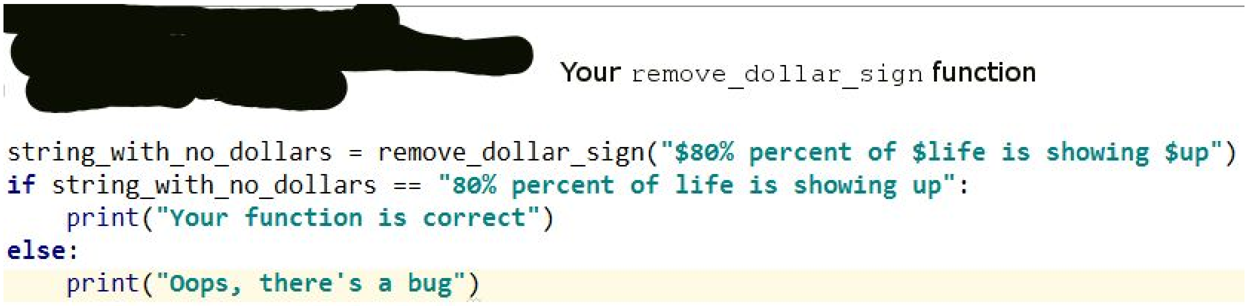
Serious exercise

5. Write a function that removes the dollar sign ("$") in a string, named `remove_dollar_sign`, takes 1 parameter: `s`, where s is the input string, returns the new string with no dollar sign in it
   Hint: Google "Python string replace remove"
6. Now, another programmer named Hiep will use your code in exercise 3. He writes as follows:

```
string_with_no_dollars = remove_dollar_sign("$80% percent of $life is to
show $up")
if string_with_no_dollars == "80% percent of life is to show up":
    print("Your function is correct")
```

```
else:
    print("Oops, there's a bug")
```

Copy this code into your editor, run the whole program and see what it prints out:



[redacted]

Your `remove_dollar_sign` function

```
string_with_no_dollars = remove_dollar_sign("$80% percent of $life is showing $up")
if string_with_no_dollars == "80% percent of life is showing up":
    print("Your function is correct")
else:
    print("Oops, there's a bug")
```

If it prints out **“Your function is correct”**, we’re good

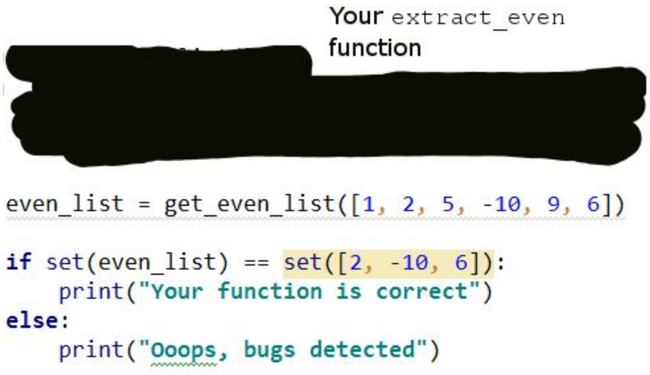
If it prints out **“Oops, there’s a bug”**, you might want to come back and check your function

7. Write a function that extracts the even items in a given integer list, named `extract_even`, takes 1 parameter: `l`, where `l` is the given integer list ([1, 4, 5, -1, 10] for example), returns a new list contains only even numbers ([4, 10] if the given list is [1,4,5,-1,10])
8. Let’s take your function to the test. The tester will write his/her test code as follows:

```
even_list = get_even_list([1, 2, 5, -10, 9, 6])

if set(even_list) == set([2, -10, 6]):
    print("Your function is correct")
else:
    print("Ooops, bugs detected")
```

Copy this code into your editor, run the whole program and see what it prints out:



Your `extract_even` function

[redacted]

```
even_list = get_even_list([1, 2, 5, -10, 9, 6])

if set(even_list) == set([2, -10, 6]):
    print("Your function is correct")
else:
    print("Ooops, bugs detected")
```

If it prints out **“Your function is correct”**, we’re good

If it prints out **“Oops, bugs detected”**, you might want to come back and check your function