Solution Review: Product of Two Positive Integers

This lesson contains the solution review for the challenge to find the product of two numbers.

WE'LL COVER THE FOLLOWING

Let's discuss the solution to the challenge in the previous lesson. The problem was to find the product of two positive integers.

Implementation

Let's have a look at the implementation below:

```
def recursive_multiply(x, y):

    # This cuts down on the total number of
    # recursive calls:
    if x < y:
        return recursive_multiply(y, x)
    if y == 0:
        return 0
    return x + recursive_multiply(x, y-1)

x = 500
y = 2000

print(x * y)
print(recursive_multiply(x, y))
```

Explanation

The hint indicated the following:

```
5 * 3 = 5 + 5 + 5 = 15
```

We make use of the hint in the implementation. Let's skip the code on **lines 5-6** for a while and discuss the code afterward. On **line** 7, we check if `y` equals `0`. If it does, `0` is returned on **line 8**. Otherwise, `x` is added to the sum returned from the recursive call on **line 9**. `y-1` is passed to the next recursive call as `x` is added once in the current recursive call. So overall, `x` will be added together `y` times in all the recursive calls. This will return the product of `x` and `y` at the end of all the recursive calls.

Now, in the implementation provided above, we make `y` recursive calls so if `x` equals `500` and `y` equals `2000`, we get the following error:

```
RecursionError: maximum recursion depth exceeded in comparison
```

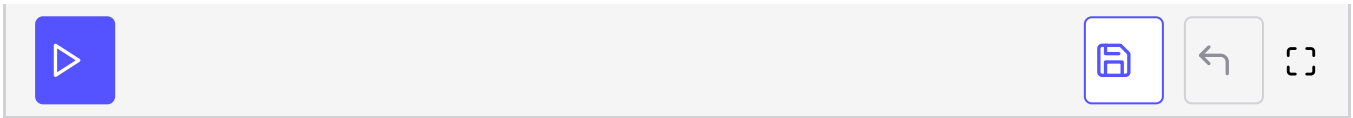
if we skip the **lines 5-6** from the above implementation. Check out the code below:

```
def recursive_multiply(x, y):

    # This cuts down on the total number of
    # recursive calls:
    if y == 0:
        return 0
    return x + recursive_multiply(x, y-1)

x = 500
y = 2000

print(x * y)
print(recursive_multiply(x, y))
```



We get `maximum recursion depth exceeded in comparison` whenever the depth of the recursion tree exceeds a limit.

Therefore, we add the following lines:

```
if x < y:
        return recursive_multiply(y, x)
```

In the code above, we swap `y` and `x` to cut down on the number of recursive

calls in case `x` is less than `y`. However, there isn't anything we can do if both

`x` and `y` are large enough to cause the `Recursion Error: maximum recursion depth exceeded in comparison`.

With this, we come to an end to the chapter on recursion. In the next chapter, we'll explore quite a few problems on string processing.