Congratulations! You passed!

Grade received 100% To pass 80% or higher

Go to next item

1. What is recursion used for? **1 / 1 point**

- ○ Recursion is used to create loops in languages where other loops are not available.
- ○ We use recursion only to implement mathematical formulas in code.
- ○ Recursion is used to iterate through sequences of files and directories.
- ◉ Recursion lets us tackle complex problems by reducing the problem to a simpler one.

Correct
You nailed it! By reducing the problem to a smaller one each time a recursive function is called, we can tackle complex problems in simple steps.

2. Which of these activities are good use cases for recursive programs? Check all that apply. **1 / 1 point**

- ☑ Going through a file system collecting information related to directories and files.

Correct
Right on! Because directories can contain subdirectories that can contain more subdirectories, going through these contents is a good use case for a recursive program.

- ☐ Creating a user account.
- ☐ Installing or upgrading software on the computer.
- ☑ Managing permissions assigned to groups inside a company, when each group can contain both subgroups and users.

Correct
You got it! As the groups can contain both groups and users, this is the kind of problem that is a great use case for a recursive solution.

- ☐ Checking if a computer is connected to the local network.

3. Fill in the blanks to make the is_power_of function return whether the number is a power of the given base. Note: base is assumed to be a positive number. Tip: for functions that return a boolean value, you can return the result of a comparison. **1 / 1 point**

```
def is_power_of(number, base):
  # Base case: when number is smaller than base.
  if number < base:
    # If number is equal to 1, it's a power (base**0).
    return number == 1

  # Recursive case: keep dividing number by base.
  return is_power_of(number // base, base)

print(is_power_of(8,2)) # Should be True
print(is_power_of(64,4)) # Should be True
print(is_power_of(70,10)) # Should be False
```

Run
Reset

```
True
True
False
```

Correct
Nice job! You've made the code check for powers of numbers by reducing the problem to a smaller one.

4. The count_users function recursively counts the amount of users that belong to a group in the company system, by going through each of the members of a group and if one of them is a group, recursively calling the function and counting the members. But it has a bug! Can you spot the problem and fix it? **1 / 1 point**

```
def count_users(group):
  count = 0
  for member in get_members(group):
    if is_group(member):
      count += count_users(member)
    else:
      count += 1
  return count

print(count_users("sales")) # Should be 3
```

```
print(count_users("engineering")) # Should be 8
print(count_users("everyone")) # Should be 18
```

Run

Reset

```
3
8
18
```

Correct

Well done, you! You spotted the problem that was causing groups to be counted when we only wanted to count users!

5. Implement the sum_positive_numbers function, as a recursive function that returns the sum of all positive numbers between the number n received and 1. For example, when n is 3 it should return 1+2+3=6, and when n is 5 it should return 1+2+3+4+5=15.

1 / 1 point

```
def sum_positive_numbers(n):
  if n==0:
    return n;
  return n + sum_positive_numbers(n-1)

print(sum_positive_numbers(3)) # Should be 6
print(sum_positive_numbers(5)) # Should be 15
```

Run

Reset

```
6
15
```

Correct

Here is your output:
6
15

Great work! You've really nailed writing recursive functions!