#Bonus challenge

**Yes, I think the following expressions are equivalent.**

**In case of not( x or y ), ‘or’ method prints True when at one of x and y is True or both of them are True. Thus, the final result will be False due to ‘method’ unless both x and y are False.**

**In addition, not x and not y, ‘and’ method prints True when both not(x) and not(y) are True, which means if any of not(x) and not(y) is False, the result will be False. To get True, same as first expression, both x and y have to be False. Therefore, expressions, not( x or y ) & not x and not y is equivalent.**

**Here are the possible combinations.**

not(x or y) -->

x = True, y = True 🡪 False

x = False, y = True 🡪 False

x = True, y = False 🡪 False

x = False, y = False 🡪 True

not x and not y -->

x = True, y = True 🡪 False

x = False, y = True 🡪 False

x = True, y = False 🡪 False

x = False, y = False 🡪 True

**[not (x and y)]: ‘and’ method prints True only if both x and y are True. Therefore, the final answer will be False if both x and y is True.**

**[not x or not y]: ‘or’ method prints False only if both x and y are True. Therefore, the final answer will be False if both x and y is True.**

**Here are the possible combinations.**

not(x and y) 🡪

x = True, y = True 🡪 False

x = False, y = True 🡪 True

x = True, y = False 🡪 True

x = False, y = False 🡪 True

not x or not y 🡪

x = True, y = True 🡪 False

x = False, y = True 🡪 True

x = True, y = False 🡪 True

x = False, y = False 🡪 True

Since all combinations are same, [not(x and y)] is equivalent to [not x or not y]