Currently I'm working on a Python programming problem. The current program below contains some bugs. Can you help in fixing these bugs with as few changes as possible? Below I first provide the problem description and then the buggy program.

Given two positive integers A and B, find GCD of A and B.

Buggy Program:

```
class P1_Python_buggy:

  def findGCD(A, B):
    while B != 0:
      temp = B
      B = A % B
      A = temp

  def main():
    A = 56
    B = 98

    gcd = P1_Python_buggy.findGCD(A, B)

    print(f"GCD of {A} and {B} is: {gcd}")


if __name__ == "__main__":
  P1_Python_buggy.main()
```

Can you fix the above buggy program? Modify only the code within the function; make sure that you make minimal possible changes needed to fix the program.

Vorschläge anzeigen

The given program is buggy because it doesn't handle the case where B becomes 0 before A. This can happen when A is a multiple of B. In such cases, the program will enter an infinite loop.

To fix this, we need to ensure that the loop terminates even when B becomes 0. We can do this by checking if B is 0 before entering the loop and returning A if it is.

Here's the corrected program:

Python

```python
class P1_Python_buggy:
    def findGCD(A, B):
        if B == 0:
            return A
        while B != 0:
            temp = B
            B = A % B
            A = temp
        return A

    def main():
        A = 56
        B = 98

        gcd = P1_Python_buggy.findGCD(A, B)

        print(f"GCD of {A} and {B} is: {gcd}")


if __name__ == "__main__":
    P1_Python_buggy.main()
```

Verwende den Code mit Vorsicht.

This modification ensures that the program correctly handles the case where B becomes 0, preventing the infinite loop.