



I'm trying to understand a given program for a Python programming problem. Can you help by explaining a specific part of this program? Below I first provide the problem description, then the program, and then a specific part of this program.

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

Program:

```
def find_gcd(A, B):
    while B != 0:
        temp = B
        B = A % B
        A = temp
    return A

def main():
    A = 56
    B = 98

    gcd = find_gcd(A, B)

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

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

Specific part:

```
while B != 0:
    temp = B
    B = A % B
    A = temp
```

Can you provide a detailed explanation about the specific part above in the context of the whole program?

Vorschläge anzeigen



I'd be glad to provide a detailed explanation of the specific part of the Python program you've shared:

Specific part:

Python

```
while B != 0:
    temp = B
    B = A % B
    A = temp
```

Verwende den Code [mit Vorsicht](#).



Explanation:

This code block implements the **Euclidean algorithm**, a classic method for finding the greatest common divisor (GCD) of two positive integers. Here's a breakdown of what each line does:

- `while B != 0:` : This loop continues as long as `B` is not equal to 0. The algorithm terminates when `B` becomes 0.
- `temp = B` : This line temporarily stores the current value of `B` in the variable `temp`. This is crucial for the subsequent calculations.
- `B = A % B` : This line calculates the remainder of `A` divided by `B` and assigns it to `B`. In essence, it replaces `B` with the remainder of the division.
- `A = temp` : This line assigns the value stored in `temp` (which was the original value of `B`) to `A`.

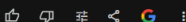
How it works:

The Euclidean algorithm leverages the following property: the GCD of two numbers `A` and `B` is also the GCD of `B` and the remainder of `A` divided by `B`.

- Initial state:** The loop starts with `A` and `B` being the original numbers.
- Iteration:**
 - The remainder of `A` divided by `B` is calculated and stored in `B`.
 - The original value of `B` (stored in `temp`) is assigned to `A`.
- Termination:** The loop continues until `B` becomes 0. At this point, `A` holds the GCD of the original `A` and `B`.

Overall purpose:

The goal of this code block is to efficiently find the GCD of `A` and `B` using the Euclidean algorithm. By iteratively calculating remainders and updating `A` and `B`, the algorithm eventually converges on the GCD.



Prompt hier eingeben

