

Please consider this programming problem:

Given an integer N , find the number of divisors of N that are divisible by 3.

Provide an example solution for the problem in the Python language.



Certainly! The problem requires us to find the number of divisors of a given integer N that are also divisible by 3.

Example Solution in Python

```
python Code kopieren

def count_divisors_divisible_by_3(N):
    # Initialize a counter to keep track of divisors divisible by 3
    count = 0

    # Iterate through all numbers from 1 to the square root of N
    for i in range(1, int(N**0.5) + 1):
        if N % i == 0: # Check if i is a divisor of N
            if i % 3 == 0: # Check if i is divisible by 3
                count += 1
            if (N // i) % 3 == 0 and i != N // i: # Check the paired divisor and ensure i
                count += 1

    return count

# Example usage
N = 18
result = count_divisors_divisible_by_3(N)
print(f"The number of divisors of {N} that are divisible by 3 is: {result}")
```

Explanation:

- Finding Divisors:** We loop through numbers from 1 to \sqrt{N} because if i is a divisor of N , then $N // i$ is also a divisor. This helps reduce the number of iterations.
- Checking for Divisibility by 3:** For each divisor i , we check if i is divisible by 3. If it is, we increment our counter. We also check the paired divisor $N // i$ (only if it's different from i) to see if it is divisible by 3.
- Edge Cases:** The solution works for all integers N , including 1, which has only one divisor (itself) and none divisible by 3.

Example:

For $N = 18$:

- Divisors of 18: 1, 2, 3, 6, 9, 18
- Divisors divisible by 3: 3, 6, 9, 18

Thus, the function returns 4 .

Output:

```
python Code kopieren

The number of divisors of 18 that are divisible by 3 is: 4
```

This solution is efficient and handles the problem within a reasonable time complexity of $O(\sqrt{N})$.

👍 🔄 🗑️ 📄 🌟



Sende eine Nachricht an ChatGPT



ChatGPT kann Fehler machen. Überprüfe wichtige Informationen.

