

4. Write a program to find the GCD of two numbers using recursive factorization

**Code:**

```
def prime_factors(n, factor=2):
    if n <= 1:
        return []
    while n % factor != 0:
        factor += 1
    return [factor] + prime_factors(n // factor, factor)

def common_factors(factors1, factors2):
    if not factors1 or not factors2:
        return []
    if factors1[0] == factors2[0]:
        return [factors1[0]] + common_factors(factors1[1:], factors2[1:])
    elif factors1[0] < factors2[0]:
        return common_factors(factors1[1:], factors2)
    else:
        return common_factors(factors1, factors2[1:])

def gcd_recursive_factorization(a, b):
    factors_a = prime_factors(a)
    factors_b = prime_factors(b)
    common = common_factors(sorted(factors_a), sorted(factors_b))
    gcd = 1
    for factor in common:
        gcd *= factor
    return gcd

num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))

result = gcd_recursive_factorization(num1, num2)
print(f"The GCD of {num1} and {num2} is: {result}")
```

**Output:**

```
Enter the first number: 32
Enter the second number: 64
The GCD of 32 and 64 is: 32
```

**Time Complexity:**

- $T(n) = O(\sqrt{n})$