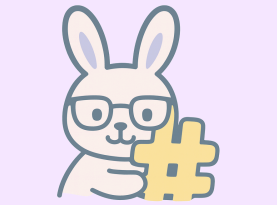
Factor Problems



Review: Modulo %

When dividing two numbers, we get a **quotient** and a **remainder**.

Example: 17 ÷ 5 = 3 remainder 2.

Python code:

```
print(17 // 5)  # quotient is 3
print(17 % 5)   # remainder is 2
```

Factor

Definition: If `n ÷ i` has zero remainder, then `i` is a factor of `n`.

Examples:

- Factors of `9` are `1, 3, 9`.
- Factors of `16` are `1, 2, 4, 8, 16`.

Important: Factors of `n` are between `1` and `n` (inclusive).

Factor

In Python, if `n % i == 0`, then `i` is a factor of `n`.

Example:

```
n = 9
i = 3

if n % i == 0:
    print(str(i) + " is a factor of " + str(n))
```

Factor

In Python, if `n % i == 0`, then `i` is a factor of `n`.

Example:

```
n = 9
i = 3

if n % i == 0:
    print(str(i) + " is a factor of " + str(n))
```

Output:

```
3 is a factor of 9
```

Find Factors

Goal: Given a positive integer `n`, print all of its factors.

Find Factors

Goal: Given a positive integer `n`, print all of its factors.

Idea:

- Try all `i` among `1, 2, ..., n`.
- If `n % i == 0`, we know `i` is a factor of `n`.

Find Factors

Python code:

```
n = 12

for i in range(1, n + 1):  # 1, 2, ..., n
    if n % i == 0:
        print(i)
```

Find Factors

Python code:

```
n = 12

for i in range(1, n + 1):  # 1, 2, ..., n
    if n % i == 0:
        print(i)
```

Output:

```
1
2
3
4
6
12
```

Prime Number

Definition: `n` is a **prime number** if it has exactly two factors, `1` and `n`.

Prime Number

Definition: `n` is a **prime number** if it has exactly two factors, `1` and `n`.

Examples:

- `1` is not prime (because it has only one factor)
- `2` is prime (factors: 1, 2)
- `5` is prime (factors: 1, 5)
- `9` is not prime (factors: 1, 3, 9)

Prime Number

Definition: `n` is a **prime number** if it has exactly two factors, `1` and `n`.

Examples:

- `1` is not prime (because it has only one factor)
- `2` is prime (factors: 1, 2)
- `5` is prime (factors: 1, 5)
- `9` is not prime (factors: 1, 3, 9)

Important: For any `i` between `2` and `n-1` (inclusive), if `i` is a factor of `n`, then `n` is not a prime.

Prime Check

Goal: Given integer `n` (at least 2), tell whether `n` is prime or not.

Prime Check

Goal: Given integer `n` (at least 2), tell whether `n` is prime or not.

Idea:

- Try all `i` among `2, 3, ..., n-1`.
- If any `n % i == 0`, we know `n` is **not** prime.

Prime Check

Python code:

```python
n = 29

is_prime = True

for i in range(2, n):  # 2, 3, ..., n-1
    if n % i == 0:
        is_prime = False
        break

print(is_prime)
```

Prime Check

Python code:

```
n = 29

is_prime = True

for i in range(2, n):  # 2, 3, ..., n-1
    if n % i == 0:
        is_prime = False
        break

print(is_prime)
```

Output:

```
True
```

Prime Check

A different example: `n = 21`.

```
n = 21

is_prime = True

for i in range(2, n):  # 2, 3, ..., n-1
    if n % i == 0:
        is_prime = False
        break

print(is_prime)
```

Prime Check

A different example: `n = 21`.

```
n = 21

is_prime = True

for i in range(2, n):  # 2, 3, ..., n-1
    if n % i == 0:
        is_prime = False
        break

print(is_prime)
```

Output:

```
False
```

Question 🖨️ Prime check

What is the output?

```
n = 21

is_prime = True

for i in range(2, n):  # 2, 3, ..., n-1
    if n % i == 0:
        print(i)  # added this line
        is_prime = False
        break

print(is_prime)
```

Question 🖨️ Prime check

What is the output?

```
n = 21

is_prime = True

for i in range(2, n):  # 2, 3, ..., n-1
    if n % i == 0:
        print(i)  # added this line
        is_prime = False
        break

print(is_prime)
```

Output:

```
3
False
```

Summary

- **Factor check:** If `n % i == 0`, then `i` is a factor of `n`.
- **Find all factors:** For `i` in `[1, ..., n]`, print `i` if `n % i == 0`.
- **Prime check:** Test if `n` has any factor in `[2, 3, ..., n-1]`.