1. Write a Python function find_smallest_multiple(n: int) that uses a while loop to find the smallest positive integer x such that x is divisible by all numbers from 1 to n.

- Conditions:

1. The function should keep incrementing x by 1 until it finds a number that satisfies the condition.
2. Your solution should handle the edge case where n = 1 efficiently, returning 1 directly since 1 is divisible by itself.

In [1]:
```python
def find_smallest_multiple(n: int) -> int:

    if n == 1:
        return 1

    x = n

    while True:
        is_divisible_by_all = True

        for i in range(1, n + 1):
            if x % i != 0:
                is_divisible_by_all = False
                break

        if is_divisible_by_all:
            return x

        x += 1

print(f"The smallest positive integer divisible by all numbers from 1 to 1 is: {find_smallest_multiple(1)}")
print(f"The smallest positive integer divisible by all numbers from 1 to 5 is: {find_smallest_multiple(5)}")
print(f"The smallest positive integer divisible by all numbers from 1 to 7 is: {find_smallest_multiple(7)}")
```

```
The smallest positive integer divisible by all numbers from 1 to 1 is: 1
The smallest positive integer divisible by all numbers from 1 to 5 is: 60
The smallest positive integer divisible by all numbers from 1 to 7 is: 420
```

2. Write a function cubesum() that accepts an integer and returns the sum of the cubes of individual digits of that number. Use this function to make functions PrintArmstrong() and isArmstrong() to print Armstrong numbers and to find whether is an Armstrong number.

In [2]:
```python
def cubesum(num: int) -> int:
    total = 0
    temp_num = num
    while temp_num > 0:
        digit = temp_num % 10
        total += digit ** 3
        temp_num //= 10
    return total

def isArmstrong(num: int) -> bool:
    return num == cubesum(num)

def PrintArmstrong(limit: int):
    print(f"Armstrong numbers (sum of cubes) up to {limit}:")
    for number in range(1, limit + 1):
        if isArmstrong(number):
            print(number, end=" ")
    print()

print(f"The cube sum of 153 is: {cubesum(153)}")
print(f"Is 153 an Armstrong number? {isArmstrong(153)}")
print(f"Is 123 an Armstrong number? {isArmstrong(123)}")
PrintArmstrong(1000)
```

```
The cube sum of 153 is: 153
Is 153 an Armstrong number? True
Is 123 an Armstrong number? False
Armstrong numbers (sum of cubes) up to 1000:
1 153 370 371 407
```

3. Why is operator precedence important? Give an example where neglecting precedence changes the result.

In [3]:
```python
# * has higher precedence than +
result_correct = 10 + 5 * 2
```

```
# 1. 5 * 2 is calculated first = 10
# 2. 10 + 10 is calculated next = 20
print(f"With precedence (correct): {result_correct}")
```

```
With precedence (correct): 20
```

In [ ]:
```
#use ( ) to force the addition to happen first
result_incorrect = (10 + 5) * 2

#1. (10 + 5) is calculated first = 15
#2. 15 * 2 is calculated next = 30
print(f"Neglecting precedence (incorrect): {result_incorrect}")
```

```
Neglecting precedence (incorrect): 30
```

4. Write a program to input a decimal number and print its equivalent binary, octal, and hexadecimal using operators.

In [5]:
```
decimal_num = int(input("Enter a decimal number: "))

binary_val = bin(decimal_num)
octal_val = oct(decimal_num)
hexadecimal_val = hex(decimal_num)

print(f"The binary equivalent is: {binary_val}")
print(f"The octal equivalent is: {octal_val}")
print(f"The hexadecimal equivalent is: {hexadecimal_val}")
```

```
The binary equivalent is: 0b110111
The octal equivalent is: 0o67
The hexadecimal equivalent is: 0x37
```

5. Write a Python function to create and print a list where the values are the squares of numbers between 1 and 30 (both included).

In [6]:
```
def create_list_of_squares():
    squares_list = []
    for i in range(1, 31):
        squares_list.append(i * i)

    print("List of squares from 1 to 30:")
    print(squares_list)
```

```
create_list_of_squares()
```

```
List of squares from 1 to 30:
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400, 441, 484, 529, 576, 625, 676, 729, 78
4, 841, 900]
```

6. Write a program that takes a sentence as input and counts the frequency of vowels, consonants, digits, and special characters separately

In [7]:
```
sentence = input("Enter a sentence: ")

vowels = 0
consonants = 0
digits = 0
specials = 0

vowel_chars = "aeiouAEIOU"

for char in sentence:
    if char in vowel_chars:
        vowels += 1
    elif 'a' <= char.lower() <= 'z':
        consonants += 1
    elif '0' <= char <= '9':
        digits += 1
    elif char != ' ' and char != '\t' and char != '\n':
        specials += 1

print("\n--- Character Frequency Report ---")
print(f"Vowels: {vowels}")
print(f"Consonants: {consonants}")
print(f"Digits: {digits}")
print(f"Special Characters: {specials}")
```

```
--- Character Frequency Report ---
Vowels: 6
Consonants: 11
Digits: 0
Special Characters: 0
```

7. Write a Python program to create a dictionary of students' names as keys and their marks as values. Then:

- Print the student with the highest marks
- Print the student with the lowest marks

In [8]:

```python
student_marks = {
    "Alice": 88,
    "Bob": 95,
    "Charlie": 72,
    "Diana": 98,
    "Ethan": 65,
    "Fiona": 95
}

if not student_marks:
    print("The dictionary is empty.")
else:
    student_list = list(student_marks.keys())

    highest_marks = student_marks[student_list[0]]
    student_with_highest = student_list[0]

    lowest_marks = student_marks[student_list[0]]
    student_with_lowest = student_list[0]

    for student, marks in student_marks.items():
        if marks > highest_marks:
            highest_marks = marks
            student_with_highest = student

        if marks < lowest_marks:
            lowest_marks = marks
            student_with_lowest = student

    print("--- Student Marks Report ---")
    print(f"Student with the highest marks: {student_with_highest} ({highest_marks})")
    print(f"Student with the lowest marks: {student_with_lowest} ({lowest_marks})")
```

```
--- Student Marks Report ---
Student with the highest marks: Diana (98)
Student with the lowest marks: Ethan (65)
```