INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI
DEPARTMENT OF MATHEMATICS AND STATISTICS

Class Test - 5 **MA517M-Basic Programming Laboratory** 15 September 2025
Name **Roll No.: MA25M**

C++ Programming Question: Heptaphobic Numbers

In number theory, a **heptaphobic number** is defined as follows:

A **positive integer** is called *heptaphobic* if:

1. It is **not divisible by 7**, and
2. **No number divisible by 7** can be produced by **swapping any two digits** of the number (without creating leading zeros).

Examples

- **17** is heptaphobic: Not divisible by 7. Swapping digits gives 71, which is not divisible by 7. ✓
- **103** is heptaphobic: Not divisible by 7. Swapping any digits (e.g., 130, 301, etc.) does not produce a number divisible by 7. ✓
- **14** is **not** heptaphobic: It is divisible by 7. ×
- **231** is **not** heptaphobic: It is not divisible by 7, but swapping digits gives 213, which **is** divisible by 7. ×

Task

Write a **C++ program** to:

1. Read an integer `N` from the user.
2. Count and print the total number of **heptaphobic numbers strictly less than** `N`.
3. Your program should:
   - Use appropriate loops and conditionals.
   - Check for divisibility by 7.
   - Generate all valid digit swaps (excluding those that produce numbers with leading zeros).
   - Ensure efficiency for reasonably large `N` (e.g., up to 10,000).

Expected Output

```
Total heptaphobic numbers less than N: <count>
```

INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI
DEPARTMENT OF MATHEMATICS AND STATISTICS

Class Test - 5 **MA517M-Basic Programming Laboratory** 15 September 2025
Name **Roll No.: MA25M**

Problem: Truncatable Primes (Advanced Prime Manipulation)

A prime number is called **truncatable** if it remains prime when digits are successively removed from either the left or the right.

Definition

A number is a **truncatable prime** if:

1. The number itself is prime.
2. All numbers obtained by successively removing digits from **left to right** are prime.
3. All numbers obtained by successively removing digits from **right to left** are prime.

Note: The single-digit primes (2, 3, 5, and 7) are **not considered** truncatable primes.

Example

Consider the number `3797`:

- **Left to right truncation:** `3797`, `797`, `97`, `7` — all are prime. ✓
- **Right to left truncation:** `3797`, `379`, `37`, `3` — all are prime. ✓

Therefore, `3797` is a **truncatable prime**.

Task

Write a program in C++ to:

1. Identify all prime numbers that are truncatable from both left to right and right to left.
2. There are exactly **eleven** such primes.
3. Compute and print the **sum** of these eleven truncatable primes.

Output Format

```
11 Truncable Prime: 23, 37, 53, 73, 313, 317, 373, 797, 3137, 3797, 739397
Sum of the eleven truncatable primes: 748317
```

INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI
DEPARTMENT OF MATHEMATICS AND STATISTICS

Class Test - 5 **MA517M-Basic Programming Laboratory** 15 September 2025

Name **Roll No.: MA25M**

Problem: Circular Primes (Advanced Prime Manipulation)

A prime number is called a **circular prime** if all rotations of its digits are also prime numbers.

Definition

A number is a **circular prime** if:

1. The number itself is prime.
2. All cyclic rotations of its digits also result in prime numbers.

For example, the number `197` is a circular prime because:

- Rotations: `197`, `971`, `719`
- All three numbers are prime.

Task

Write a program in C++ to:

1. Find all the circular prime numbers less than `1,000`.
2. Count the total number of such circular primes.
3. Display the count.

Constraints

- You may use digit manipulation or string rotation to generate the circular permutations.
- Ensure that the prime checking is efficient for values up to one million.

Expected Output

```
Total number of circular primes below 1,000: <count>
```

Example

Below 100, the circular primes are:

$$2,\ 3,\ 5,\ 7,\ 11,\ 13,\ 17,\ 31,\ 37,\ 71,\ 73,\ 79,\ 97$$

Total: 13 circular primes under 100.

INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI
DEPARTMENT OF MATHEMATICS AND STATISTICS

Class Test - 5 **MA517M-Basic Programming Laboratory** 15 September 2025

Name **Roll No.: MA25M**

Problem: Semiprime Count Below a Limit

A **semiprime** is a natural number that is the product of two prime numbers. These two primes may be the same (in the case of a square of a prime) or different.

Definition

A number n is called a **semiprime** if:

$$n = p \times q \quad \text{where } p \text{ and } q \text{ are prime numbers, and } p \leq q.$$

For example:

- $15 = 3 \times 5$ — semiprime
- $49 = 7 \times 7$ — semiprime
- $30 = 2 \times 3 \times 5$ — **not** a semiprime (three prime factors)

Task

Write a C++ program to:

1. Count the number of positive integers less than a given number N that have exactly **two prime factors**.
2. That is, count all **semiprimes** less than N.

Input

No input is required. Use a constant value for $N = 10^4$ in your program.

Output Format

```
Number of semiprimes less than 10000: <count>
```

Example

For $N = 100$, the semiprimes are: Total = 34 semiprimes under 100.

INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI
DEPARTMENT OF MATHEMATICS AND STATISTICS

Class Test - 5 **MA517M-Basic Programming Laboratory** 15 September 2025
Name **Roll No.: MA25M**

Problem: Prime Subset Sums

Let S be the set of the first n prime numbers. We are interested in the number of subsets of S whose elements sum to a prime number.

Definition

Let:

$$S = \{p_1, p_2, p_3, \ldots, p_n\}$$

where p_i is the i-th prime number.

A subset $A \subseteq S$ is considered valid if:

1. The sum of all elements in A is a prime number.
2. Subsets can include any number of elements

Task

Write a program in C++ to:

1. Generate the first $n = 20$ prime numbers.
2. Find all possible subsets of these primes.
3. Count how many of these subsets sum to a prime number.

Constraints

- The sum of the first 500 primes is 824693.
- The total number of subsets is 2^{500}, but you do not need to enumerate all explicitly.

Output Format

```
Number of prime-summing subsets of the first 20 primes: <count>
```

Example (Simplified)

Let $n = 5$. First 5 primes: $\{2, 3, 5, 7, 11\}$

Possible subsets whose sum is prime:

- {2}, {3}, {5}, {7}, {11}, {2,3}, {2,5}, {3,5}, {2,11}, ...
- {2,7} is not included as the sum 9 is not a prime.

INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI
DEPARTMENT OF MATHEMATICS AND STATISTICS

Class Test - 5 **MA517M-Basic Programming Laboratory** 15 September 2025
Name **Roll No.: MA25M**

Problem: Longest Collatz Sequence

The Collatz sequence for a given positive integer n is defined as follows:

- If n is even, the next term is $n/2$.
- If n is odd, the next term is $3n + 1$.
- The sequence continues until it reaches the value 1.

For example, starting with 13, the sequence is:

$$13 \to 40 \to 20 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1$$

which has 10 terms.

Task:
Write a C++ program to find the starting number under one million that produces the longest Collatz sequence (using while and if).

Your program should:

1. Iterate through all starting numbers from 1 up to (but not including) 1,000.
2. For each starting number, generate the Collatz sequence and count the number of terms.
3. Keep track of the starting number that produces the longest sequence.
4. Output the starting number with the longest Collatz sequence and the length of that sequence.

Requirements:

- Use appropriate loops, pointers, arrays and conditionals.
- Implement the logic using if and while if you wish.
- Optimize your program to run efficiently (e.g., consider caching lengths of sequences to avoid recomputation).
- Properly format your output.

Example Output:
If the upper limit is set to 14, the program output should be:

```
Enter the upper limit: 14
Starting number with longest Collatz sequence under 14 is 9
Length of the sequence: 20
```

INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI
DEPARTMENT OF MATHEMATICS AND STATISTICS

Class Test - 5 **MA517M-Basic Programming Laboratory** 15 September 2025
Name **Roll No.: MA25M**

Problem: Curious Numbers (Digit Factorials)

A number is called a **curious number** if it is equal to the sum of the factorials of its digits.

Definition

A number n is curious if:

$$n = \sum_{\text{digits } d \text{ of } n} d!$$

For example, 145 is a curious number because:

$$1! + 4! + 5! = 1 + 24 + 120 = 145$$

Task

Write a program in C++ to:

1. Find all curious numbers < 50000.
2. Calculate the sum of all such curious numbers (excluding 1 and 2, which are not considered sums).
3. Display the result.

Expected Output

```
Sum of all curious numbers is: <sum>
```

Example

Known curious numbers include:

$$145 \quad \text{since } 1! + 4! + 5! = 145$$

and

$$40585 \quad \text{since } 4! + 0! + 5! + 8! + 5! = 40585$$