📜 Strong Numbers Less Than n

Problem Definition

A **Strong Number** (also called a **Factorial Digit Sum Number** or **Factorion**) is a number that is equal to the sum of the factorials of its digits.

Examples

- `1 = 1!` ✅
- `2 = 2!` ✅
- `145 = 1! + 4! + 5! = 1 + 24 + 120 = 145` ✅
- `123 = 1! + 2! + 3! = 1 + 2 + 6 = 9` ❌

Task

Given an integer `n`, find all Strong Numbers that are **strictly less than** `n`.

Input Format

- A single integer `n` where `1 ≤ n ≤` 10^6

Output Format

- Print all Strong Numbers less than `n` in ascending order, separated by spaces
- If none exist, print nothing (empty output)

Sample Test Cases

Example 1

Input:

```
500
```

Output:

```
1 2 145
```

Explanation:

- `1` and `2` are trivially strong numbers
- `145 = 1! + 4! + 5! = 1 + 24 + 120 = 145` ✅
- No other number below 500 satisfies the property

Example 2

Input:

```
3
```

Output:

```
1 2
```

Explanation:

- Only `1` and `2` are strong numbers less than 3

Constraints

- `1 ≤ n ≤ 1,000,000`
- Time complexity should be reasonable for the given constraints
- Space complexity should be optimized

Sample Solution

```
c
```

```c
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    for(int i = 1; i < n; i++) {
        int s = 0;
        for(int j = i; j > 0; j = j / 10) {
            int t = 1;
            for(int k = 1; k <= j % 10; k++) {
                t = t * k;
            }
            s = s + t;
        }
        if(s == i)
            printf("%d ", i);
    }

    return 0;
}
```

Note: The original code had a small bug - it used `i <= n` but the problem asks for numbers **strictly less than** `n`, so it should be `i < n`. The corrected version above fixes this issue.