

### Problem 1.1 Super Numbers

A positive integer ( $>1$ ) is said to be **super** number if sum of its divisors (excluding itself) is equal to the number itself.

**Example1:** 6 is a super number. Because its divisors are: 1,2,3 and  $1+2+3=6$

**Example2:** 8 is not a super number. Because its divisors are: 1,2,4 and  $1+2+4 \neq 8$ .

Write a complete C program that will find and output all the super numbers in the range  $[2, 1000]$ .

---

### Problem 1.2 Converting Input Roman Number to Decimal

Write a complete program in C which will input a roman number from the range  $[i, xxx]$  and output the equivalent decimal number.

**Example 1:**

Input:  $ii$   
Output:  $2$

**Example 2:**

Input:  $xix$   
Output:  $19$

**Example 3:**

Input:  $xxvi$   
Output:  $26$

---

### Problem 1.3 Cube Free Numbers

An integer is said to be **cube free** if it is not divisible by the cube of any prime.

For example, 1, 2, 3, 4, 12, 20 are few cube free numbers. And the numbers 8, 24, 125 are not cube free.

Write a complete C program which will input a positive integer  $n$  and output two positive integers  $a, b$  satisfying the following equation:

$$n = a^3b$$

where  $b$  is cube free.

**Example1:**    Input:    52

	Output:	a=1, b=52
<b>Example2:</b>	Input:	100000
	Output:	a=10, b=100
<b>Example3:</b>	Input:	72
	Output:	a=2, b=9
<b>Example4:</b>	Input:	64
	Output:	a=4, b=1
<b>Example5:</b>	Input:	32
	Output:	a=2, b=4

---

#### Problem 1.4      Decomposing an Integer into Sum of Different Integers

Consider the following result:

Corresponding to any given positive integer  $n$ , one can always find a set of different integers  $i_1, i_2, i_3, \dots, i_k$  satisfying the following equation:

$$n = i_1 + i_2 + i_3 + \dots + i_k$$

Write a complete C program, which will input a positive integer  $n \leq 20$  and decompose it as mentioned above and display all possible such distinct decompositions.

<b>Example1:</b>	Input:	6
	Output:	6
		5+1
		4+2
		3+2+1
<b>Example2:</b>	Input:	13
	Output:	13
		12 + 1
		11 + 2
		10 + 3
		9 + 4
		8 + 5
		7 + 6
		10 + 2 + 1
		9 + 3 + 1
		8 + 4 + 1
		8 + 3 + 2
		7 + 5 + 1
		7 + 4 + 2

6 + 5 + 2  
6 + 4 + 3  
7 + 3 + 2 + 1  
6 + 4 + 2 + 1  
5 + 4 + 3 + 1

Note: Consider example2, in that we have not written 1+12 as already we have written 12+ 1, i.e. you should not duplicate same set of numbers.

---

### Problem 1.5      Permutations of Three Digit Integers

Write a C program to find all permutations of an input three-digit, positive integer that is input by the user. For example, if the user input is 432, the permutations are 234, 243, 342, 324, 423.