**Task 1 : Factorials**

**Description**

Most people are familiar with the factorial operator in math. 5! yields 120 because factorial means "multiply successive terms where each are one less than the previous":

5! -> 5 \* 4 \* 3 \* 2 \* 1 -> 120

Now let's reverse it. Could you write a function that tells us that "120" is "5!"?

**Hint: divide the term by successively larger terms until you get to "1" as the resultant:**

120 -> 120/2 -> 60/3 -> 20/4 -> 5/5 -> 1 => 5!

**Sample Input**

You'll be given a single integer, one per line. Examples:

120

150

**Sample Output**

Your program should report what each number is as a factorial, or "NONE" if it's not legitimately a factorial. Examples:

120 = 5!

150   NONE

**Challenge Input**

3628800

479001600

6

18​

# Doggo Competition

## Description

Your job is to create a program that lists all places within the range of 0-100 in spoken English, excluding the placing (X) of your winning pup.

## Input description

Input is the integer placement of your dog (X) within the range 0-100.

## Output description

A reader should see a neatly formatted list of placements from 0-100 in spoken English, excluding your dog's placement.

## Example of 1st Place finish

2nd, 3rd, 4th, 5th, 6th, 7th, 8th, 9th, 10th, 11th, 12th, 13th, 14th, 15th, 16th, 17th, 18th, 19th, 20th, 21st, 22nd, 23rd, 24th, 25th, 26th, 27th, 28th, 29th, 30th, 31st, 32nd, 33rd, 34th, 35th, 36th, 37th, 38th, 39th, 40th, 41st, 42nd, 43rd, 44th, 45th, 46th, 47th, 48th, 49th, 50th, 51st, 52nd, 53rd, 54th, 55th, 56th, 57th, 58th, 59th, 60th, 61st, 62nd, 63rd, 64th, 65th, 66th, 67th, 68th, 69th, 70th, 71st, 72nd, 73rd, 74th, 75th, 76th, 77th, 78th, 79th, 80th, 81st, 82nd, 83rd, 84th, 85th, 86th, 87th, 88th, 89th, 90th, 91st, 92nd, 93rd, 94th, 95th, 96th, 97th, 98th, 99th, 100th

# Goldilocks’ Bear Necessities

## Description

The task falls to you: given descriptions of Goldilocks' needs and of the available porridge/chairs at the dinner table, tell Goldilocks which chair to sit in so the chair does not break, and the porridge is at an edible temperature.

## Formal Input

The input begins with a line specifying Goldilocks' weight (as an integer in arbitrary weight-units) and the maximum temperature of porridge she will tolerate (again as an arbitrary-unit integer). This line is then followed by some number of lines, specifying a chair's weight capacity, and the temperature of the porridge in front of it.

## **Sample input:**

100 80

30 50

130 60

90 60

150 85

120 70

200 200

110 100

Interpreting this, Goldilocks has a weight of 100 and a maximum porridge temperature of 80. The first seat at the table has a chair with a capacity of 30 and a portion of porridge with the temperature of 50. The second has a capacity of 130 and temperature of 60, etc.

## Formal Output

The output must contain the numbers of the seats that Goldilocks can sit down at and eat up. This number counts up from 1 as the first seat.

## **Sample output:**

2 5

Seats #2 and #5 have both good enough chairs to not collapse under Goldilocks, and porridge that is cool enough for her to eat.

## Challenge Input

100 120

297 90

66 110

257 113

276 191

280 129

219 163

254 193

86 153

206 107

71 137

94 40

238 127

52 146

129 197

144 59

157 124

210 59

110 54

268 119

261 121

12 189

186 108

174 21

112 18

54 90

174 52

16 129

59 181

290 123

248 132

Advanced

* Model the porridge and chair as different objects
* As well as Porridge and chair at each position, there is also now a light.
* Each light has a luminosity, 0 – 100.
* If the luminosity of a position is below 70, goldilocks cannot eat at this place.

## Challenge Input

100 120

297 90 54

66 110 78

257 113 98

276 191 12

280 129 22

219 163 12

254 193 45

86 153 65

206 107 71

71 137 87

94 40 98

238 127 45

52 146 98

129 197 10

144 59 100

157 124 12

210 59 54

110 54 98

268 119 87

261 121 32

12 189 65

186 108 97

174 21 64

112 18 31

54 90 54

174 52 65

16 129 87

59 181 98

290 123 65

248 132 21

# Calculator

Create a program that will take a series of numbers (5, 3, 15), and find how those numbers can add, subtract, multiply, or divide in various ways to relate to each other. This string of numbers should result in 5 \* 3 = 15, or 15 /3 = 5, or 15/5 = 3. When you are done, test your numbers with the following strings:

4, 2, 8

6, 2, 12

6, 2, 3

9, 12, 108

4, 16, 64

For extra credit, have the program list all possible combinations.

for even more extra credit, allow the program to deal with strings of greater than three numbers. For example, an input of (3, 5, 5, 3) would be 3 \* 5 = 15, 15/5 = 3. When you are finished, test them with the following strings.

2, 4, 6, 3

1, 1, 2, 3

4, 4, 3, 4

8, 4, 3, 6

9, 3, 1, 7