USP-ICMC-SCC - Ciências de Computação

Substitutiva da Prova 2 - 16 a 19/11/2010

Algoritmos Avançados - SCC-0210

Prof. João Luís - Monitores: Roberto de Medeiros (PAE) e Raphael Ferras

Obs.: A prova é individual. O Boca estará aberto entre 16 e 19/11, das 19 às 23h00.

(3) 1. **Prime Factors**: An integer $g > 1$ is said to be *prime* if and only if its only positive divisors are itself and one (otherwise it is said to be composite). For example, the number 21 is composite; the number 23 is prime. Note that the decomposition of a positive number g into its prime factors, i.e.,

$$g = f_1 \times f_2 \times \cdots \times f_n$$

is unique if we assert that $f_i > 1$ for all i and $f_i \leq f_j$ for $i < j$.

One interesting class of prime numbers are the so-called *Mersenne* primes which are of the form $2^p - 1$. Euler proved that $2^{31} - 1$ is prime in 1772 – all without the aid of a computer.

- **Input**: The input will consist of a sequence of numbers. Each line of input will contain one number g in the range $-2^{31} < g < 2^{31}$, but different of -1 and 1. The end of input will be indicated by an input line having a value of zero.
- **Output**: For each line of input, your program should print a line of output consisting of the input number and its prime factors. For an input number $g > 0, g = f_1 \times f_2 \times \cdots \times f_n$, where each fi is a prime number greater than unity (with $f_i \leq f_j$ for $i < j$), the format of the output line should be

  $$g \texttt{ = } f_1 \texttt{ x } f_2 \texttt{ x ... x } f_n$$

  When $g < 0$, if $\mid g \mid = f_1 \times f_2 \times \cdots \times f_n$, the format of the output line should be

  $$g \texttt{ = -1 x } f_1 \texttt{ x } f_2 \texttt{ x ... x } f_n$$

- **Sample Input**

```
-190
-191
-192
-193
-194
195
196
197
198
199
200
0
```

- **Sample Output**

```
-190 = -1 x 2 x 5 x 19
-191 = -1 x 191
-192 = -1 x 2 x 2 x 2 x 2 x 2 x 2 x 3
-193 = -1 x 193
-194 = -1 x 2 x 97
195 = 3 x 5 x 13
196 = 2 x 2 x 7 x 7
197 = 197
198 = 2 x 3 x 3 x 11
199 = 199
200 = 2 x 2 x 2 x 5 x 5
```

(3) 2. **LC-Display**: A friend of you has just bought a new computer. Until now, the most powerful computer he ever used has been a pocket calculator. Now, looking at his new computer, he is a bit disappointed, because he liked the LC-display of his calculator so much. So you decide to write a program that displays numbers in an LC-display-like style on his computer.

- **Input**: The input file contains several lines, one for each number to be displayed. Each line contains two integers s, n ($1 \leq s \leq 10, 0 \leq n \leq 99\,999\,999$), where n is the number to be displayed and s is the size in which it shall be displayed.

  The input file will be terminated by a line containing two zeros. This line should not be processed.

- **Output**: Output the numbers given in the input file in an LC-display-style using s "-" signs for the horizontal segments and s "|" signs for the vertical ones. Each digit occupies exactly $s + 2$ columns and $2s + 3$ rows. (Be sure to fill all the white space occupied by the digits with blanks, also for the last digit.) There has to be exactly one column of blanks between two digits.

  Output a blank line after each number. (You will find a sample of each digit in the sample output.)

- **Sample Input**

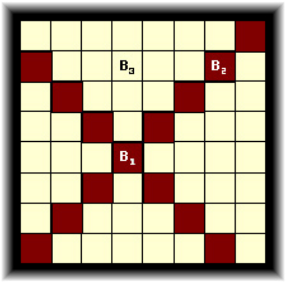
```
2 12345
3 67890
0 0
```

- **Sample Output**

```
      --   --        -- 
   |    |    | |  | |   
   |    |    | |  | |   
      --   --   --   -- 
   | |       |    |    |
   | |       |    |    |
      --   --        -- 

 ---   ---   ---   ---   --- 
|         | |   | |   | |   |
|         | |   | |   | |   |
|         | |   | |   | |   |
 ---         ---   ---       
|   |     | |   |     | |   |
|   |     | |   |     | |   |
|   |     | |   |     | |   |
 ---         ---   ---   --- 
```

(4) 3. **Little Bishops**: A bishop is a piece used in the game of chess which is played on a board of square grids. A bishop can only move diagonally from its current position and two bishops attack each other if one is on the path of the other. In the following figure, the dark squares represent the reachable locations for bishop B_1 form its current position. The figure also shows that the bishops B_1 and B_2 are in attacking positions whereas B_1 and B_3 are not. B_2 and B_3 are also in non-attacking positions.



Now, given two numbers n and k, your job is to determine the number of ways one can put k bishops on an $n \times n$ chessboard so that no two of them are in attacking positions.

- **Input**: The input file may contain multiple test cases. Each test case occupies a single line in the input file and contains two integers n ($1 \leq n \leq 8$) and k ($0 \leq k \leq n^2$).

  A test case containing two zeros for n and k terminates the input and you won't need to process this particular input.
- **Output**: For each test case in the input print a line containing the total number of ways one can put the given number of bishops on a chessboard of the given size so that no two of them are in attacking positions. You may safely assume that this number will be less than 10^{15}.
- **Sample Input**

```
8 6
4 4
0 0
```

- **Sample Output**

```
5599888
260
```