

- · Convert the decimal number 23 to binary number:  $23^{10} = 2^4 + 2^2 + 2^1 + 2^0 = (10111)_2$ .
- $\cdot$   $\;$  The value of the  $4^{th}$  index from the right in the binary representation is 0.

#### **Function Description**

Complete the function fourthBit in the editor below.

fourthBit has the following parameter(s):

int number: a decimal integer

int: an integer 0 or 1 matching the 4th least significant digit in the binary representation of number.

 $0 \le \text{number} < 2^{31}$ 

#### Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The only line contains an integer, number.

## Sample Case 0

STDIN Function 32 → number = 32

#### Sample Output 0

#### Explanation 0

- Convert the decimal number 32 to binary number: 32<sub>10</sub> = (100000)<sub>2</sub>.
- The value of the 4th index from the right in the binary representation is 0.

#### Sample Case 1

#### Sample Input 1

```
Sample Input 1

STDIN Function

77 → number = 77

Sample Output 1

1

Explanation 1

Convert the decimal number 77 to binary number: 77<sub>10</sub> = (1001101)<sub>2</sub>.

The value of the 4th index from the right in the binary representation is 1.

Answer: (penalty regime: 0 %)

Reset answer

1 | V* | * Complete the 'fourthBit' function below. * C
```

REC-CIS

|          | Test                        | Expected | Got |   |
|----------|-----------------------------|----------|-----|---|
| <b>~</b> | printf("%d", fourthBit(32)) | 0        | 0   | ~ |
| <b>~</b> | printf("%d", fourthBit(77)) | 1        | 1   | ~ |





Determine the factors of a number (i.e., all positive integer values that evenly divide into a number) and then return the  $p^{th}$  element of the list, sorted ascending. If there is no  $p^{th}$  element, return 0.

#### Example

n = 20 p = 3

The factors of 20 in ascending order are  $\{1, 2, 4, 5, 10, 20\}$ . Using 1-based indexing, if p = 3, then 4 is returned. If p > 6, 0 would be returned.

#### Function Description

Complete the function pthFactor in the editor below.

#### REC-CIS

pthFactor has the following parameter(s):

int n: the integer whose factors are to be found

int p: the index of the factor to be returned

#### Returns:

int: the long integer value of the  $p^{th}$  integer factor of n or, if there is no factor at that index, then 0 is returned

#### Constraints

 $1 \le n \le 10^{15}$  $1 \le p \le 10^9$ 

Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer n, the number to factor.

The second line contains an integer p, the 1-based index of the factor to return.

#### Sample Case 0 Sample Input 0

STDIN Function  $10 \rightarrow n = 10$  p = 3

### Sample Output 0

5

### Explanation 0

Factoring n = 10 results in {1, 2, 5, 10}. Return the  $p = 3^{rd}$  factor, 5, as the answer.

#### Sample Case 1 Sample Input 1

STDIN Function

```
10 \rightarrow n = 10

5 \rightarrow p = 5

Sample Output 1

0

Explanation 1

Factoring n = 10 results in {1, 2, 5, 10}. There are only 4 factors and p = 5, therefore 0 is returned as the answer.

Sample Case 2

Sample Input 2

STDIN Function

1 \rightarrow n = 1

1 \rightarrow p = 1
```

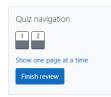
### Sample Output 2

1

### Explanation 2

Factoring n = 1 results in {1}. The p = 1st factor of 1 is returned as the answer.







Output

For each test case, print a single line containing the string "1" if you can make exactly N rupees or "0" otherwise.

SAMPLE INPUT

1

SAMPLE OUTPUT

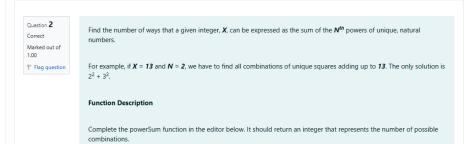
2

SAMPLE OUTPUT

0







Input Format

The first line contains an integer X.
The second line contains an integer N.

Constraints  $1 \le X \le 1000$   $2 \le N \le 10$ Output Format

Output a single integer, the number of possible combinations calculated.

Sample Input 0

powerSum has the following parameter(s):

N: the integer power to raise numbers to

X: the integer to sum to

Input Format

Sample Output 0

This is the only way in which 10 can be expressed as the sum of unique squares.

Sample Input 1

10

Sample Output 1

NEC CO

```
Explanation 1

100 = (10<sup>2</sup>) = (6<sup>2</sup> + 8<sup>2</sup>) = (1<sup>2</sup> + 3<sup>2</sup> + 4<sup>2</sup> + 5<sup>2</sup> + 7<sup>2</sup>)

Sample Input 2

100

3

Sample Output 2

1

Explanation 2

100 can be expressed as the sum of the cubes of 1, 2, 3, 4.

(1 + 8 + 27 + 64 = 100). There is no other way to express 100 as the sum of cubes.
```

```
Test Expected Got

rintf("%d", powerSum(10, 1, 2)) 1 1 V

Passed all tests! 

Finish review
```

→ Previous page