Problem 668: Kth Smallest Number in Multiplication Table

Problem Information

Difficulty: Hard
Acceptance Rate: 0.00%
Paid Only: No

Problem Description

Nearly everyone has used the

Multiplication Table

. The multiplication table of size

m x n

is an integer matrix

mat

where

mat[i][j] == i * j

(

1-indexed

).

Given three integers

m

,

n

, and

k

, return

the

k

th

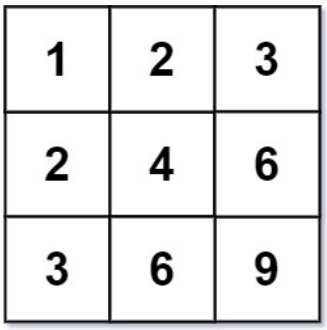
smallest element in the

m x n

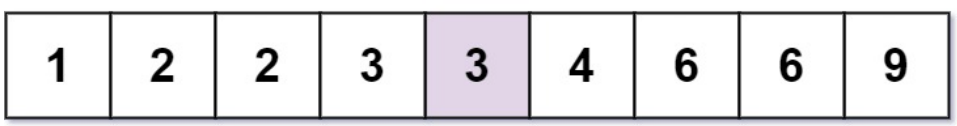
multiplication table

.

Example 1:



1	2	3
2	4	6
3	6	9



1	2	2	3	3	4	6	6	9

Input:

m = 3, n = 3, k = 5

Output:

3

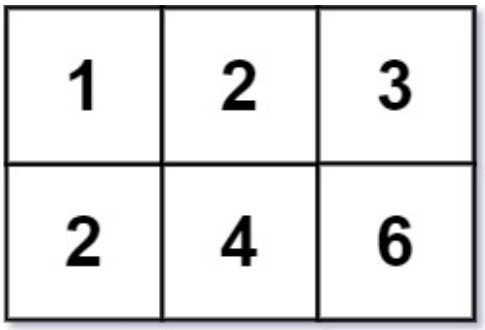
Explanation:

The 5

th

smallest number is 3.

Example 2:



Input:

m = 2, n = 3, k = 6

Output:

6

Explanation:

The 6

th

smallest number is 6.

Constraints:

1 <= m, n <= 3 * 10

4

1 <= k <= m * n

Code Snippets

C++:

```
class Solution {
public:
int findKthNumber(int m, int n, int k) {

}
};
```

Java:

```
class Solution {
public int findKthNumber(int m, int n, int k) {

}
}
```

Python3:

```
class Solution:
def findKthNumber(self, m: int, n: int, k: int) -> int:
```

Python:

```
class Solution(object):
def findKthNumber(self, m, n, k):
"""
:type m: int
:type n: int
:type k: int
:rtype: int
"""
```

JavaScript:

```
/**
* @param {number} m
* @param {number} n
* @param {number} k
* @return {number}
*/
var findKthNumber = function(m, n, k) {

};
```

TypeScript:

```
function findKthNumber(m: number, n: number, k: number): number {

};
```

C#:

```
public class Solution {
public int FindKthNumber(int m, int n, int k) {

}
}
```

C:

```
int findKthNumber(int m, int n, int k) {

}
```

Go:

```
func findKthNumber(m int, n int, k int) int {

}
```

Kotlin:

```
class Solution {
fun findKthNumber(m: Int, n: Int, k: Int): Int {

}
}
```

Swift:

```
class Solution {
func findKthNumber(_ m: Int, _ n: Int, _ k: Int) -> Int {

}
}
```

Rust:

```
impl Solution {
pub fn find_kth_number(m: i32, n: i32, k: i32) -> i32 {

}
}
```

Ruby:

```
# @param {Integer} m
# @param {Integer} n
# @param {Integer} k
# @return {Integer}
def find_kth_number(m, n, k)

end
```

PHP:

```
class Solution {

/**
```

```
* @param Integer $m
* @param Integer $n
* @param Integer $k
* @return Integer
*/
function findKthNumber($m, $n, $k) {

}
}
```

Dart:

```
class Solution {
int findKthNumber(int m, int n, int k) {

}
}
```

Scala:

```
object Solution {
def findKthNumber(m: Int, n: Int, k: Int): Int = {

}
}
```

Elixir:

```
defmodule Solution do
@spec find_kth_number(m :: integer, n :: integer, k :: integer) :: integer
def find_kth_number(m, n, k) do

end
end
```

Erlang:

```
-spec find_kth_number(M :: integer(), N :: integer(), K :: integer()) ->
integer().
find_kth_number(M, N, K) ->
.
```

Racket:

```
(define/contract (find-kth-number m n k)
(-> exact-integer? exact-integer? exact-integer? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Kth Smallest Number in Multiplication Table
* Difficulty: Hard
* Tags: math, search
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

class Solution {
public:
int findKthNumber(int m, int n, int k) {

}
};
```

Java Solution:

```
/**
* Problem: Kth Smallest Number in Multiplication Table
* Difficulty: Hard
* Tags: math, search
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

class Solution {
public int findKthNumber(int m, int n, int k) {
```

```
}
}
```

Python3 Solution:

```
"""
Problem: Kth Smallest Number in Multiplication Table
Difficulty: Hard
Tags: math, search

Approach: Optimized algorithm based on problem constraints
Time Complexity: O(n) to O(n^2) depending on approach
Space Complexity: O(1) to O(n) depending on approach
"""

class Solution:
def findKthNumber(self, m: int, n: int, k: int) -> int:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def findKthNumber(self, m, n, k):
"""
:type m: int
:type n: int
:type k: int
:rtype: int
"""
```

JavaScript Solution:

```
/**
* Problem: Kth Smallest Number in Multiplication Table
* Difficulty: Hard
* Tags: math, search
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
```

```
* Space Complexity: O(1) to O(n) depending on approach
*/

/**
* @param {number} m
* @param {number} n
* @param {number} k
* @return {number}
*/
var findKthNumber = function(m, n, k) {

};
```

TypeScript Solution:

```
/**
* Problem: Kth Smallest Number in Multiplication Table
* Difficulty: Hard
* Tags: math, search
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

function findKthNumber(m: number, n: number, k: number): number {

};
```

C# Solution:

```
/*
* Problem: Kth Smallest Number in Multiplication Table
* Difficulty: Hard
* Tags: math, search
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/
```

```
public class Solution {
public int FindKthNumber(int m, int n, int k) {

}
}
```

C Solution:

```
/*
* Problem: Kth Smallest Number in Multiplication Table
* Difficulty: Hard
* Tags: math, search
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

int findKthNumber(int m, int n, int k) {

}
```

Go Solution:

```
// Problem: Kth Smallest Number in Multiplication Table
// Difficulty: Hard
// Tags: math, search
//
// Approach: Optimized algorithm based on problem constraints
// Time Complexity: O(n) to O(n^2) depending on approach
// Space Complexity: O(1) to O(n) depending on approach

func findKthNumber(m int, n int, k int) int {

}
```

Kotlin Solution:

```
class Solution {
fun findKthNumber(m: Int, n: Int, k: Int): Int {
```

```
}
}
```

Swift Solution:

```
class Solution {
func findKthNumber(_ m: Int, _ n: Int, _ k: Int) -> Int {

}
}
```

Rust Solution:

```
// Problem: Kth Smallest Number in Multiplication Table
// Difficulty: Hard
// Tags: math, search
//
// Approach: Optimized algorithm based on problem constraints
// Time Complexity: O(n) to O(n^2) depending on approach
// Space Complexity: O(1) to O(n) depending on approach

impl Solution {
pub fn find_kth_number(m: i32, n: i32, k: i32) -> i32 {

}
}
```

Ruby Solution:

```
# @param {Integer} m
# @param {Integer} n
# @param {Integer} k
# @return {Integer}
def find_kth_number(m, n, k)

end
```

PHP Solution:

```
class Solution {
```

```
/**
* @param Integer $m
* @param Integer $n
* @param Integer $k
* @return Integer
*/
function findKthNumber($m, $n, $k) {

}
}
```

Dart Solution:

```
class Solution {
int findKthNumber(int m, int n, int k) {

}
}
```

Scala Solution:

```
object Solution {
def findKthNumber(m: Int, n: Int, k: Int): Int = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec find_kth_number(m :: integer, n :: integer, k :: integer) :: integer
def find_kth_number(m, n, k) do

end
end
```

Erlang Solution:

```
-spec find_kth_number(M :: integer(), N :: integer(), K :: integer()) ->
integer().
find_kth_number(M, N, K) ->
```

```
.
```

Racket Solution:

```
(define/contract (find-kth-number m n k)
(-> exact-integer? exact-integer? exact-integer? exact-integer?)
)
```