Ex. No. : 5.6 **Date:**

Register No: 231501049 **Name: GNAANESH B B**

Find the Factor

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.

Constraints

$1 \leq n \leq 10^{15}$

$1 \leq p \leq 10^{9}$

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
10
3
Sample Output 0
5
Explanation 0
Factoring n = 10 results in {1, 2, 5, 10}. Return the p = 3rd factor, 5, as the answer.
Sample Case 1
Sample Input 1
10
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
1
1
Sample Output 2
1
Explanation 2
Factoring n = 1 results in {1}. The p = 1st factor of 1 is returned as the answer.

For example:

Input	Result
10 3	5
10 5	0
1 1	1

PROGRAM:

```
import sys
import math


def find_factors(n):
    factors = []
    for i in range(1, int(math.sqrt(n)) + 1):
    if n % i == 0:
    factors.append(i)
    if i != n // i:
        factors.append(n // i)
    return sorted(factors)


def get_pth_factor(n, p):
    factors = find_factors(n)
    if p <= len(factors):
    return factors[p - 1]
```

```
    else:

    return 0


# Reading input directly from the standard input (typically for competitive programming)

input = sys.stdin.read

data = input().split()

n = int(data[0])

p = int(data[1])


# Calculate and print the p-th factor

print(get_pth_factor(n, p))
```

	Input	Expected	Got	
✔	10 3	5	5	✔
✔	10 5	0	0	✔
✔	1 1	1	1	✔

Passed all tests! ✔

Ex. No. : 5.7 **Date:**

Register No: 231501049 **Name: GNAANESH B B**

Merge List

Write a Python program to Zip two given lists of lists.

Input:
m : row size
n: column size
list1 and list 2 : Two lists
Output
Zipped List : List which combined both list1 and list2

Sample test case

Sample input
2
2
1
3
5
7
2
4
6
8

Sample Output

[[1, 3, 2, 4], [5, 7, 6, 8]]

PROGRAM:

```
def zip_lists(list1, list2):
    return [row1 + row2 for row1, row2 in zip(list1, list2)]


def main():
```

```
    m = int(input())
    n = int(input())


    list1 = [[int(input()) for _ in range(n)] for _ in range(m)]
    list2 = [[int(input()) for _ in range(n)] for _ in range(m)]

    zipped_list = zip_lists(list1, list2)
    print(zipped_list)

if __name__ == "__main__":
    main()
```

	Input	Expected
✔	2 2 1 2 3 4 5 6 7 8	[[1, 2, 5, 6], [3, 4, 7, 8]]

Passed all tests! ✔

Ex. No. : 5.8 **Date:**

Register No: 231501049 **Name: GNAANESH B B**

Merge Two Sorted Arrays Without Duplication

Output is a merged array without duplicates.

Input Format
N1 - no of elements in array 1
Array elements for array 1
N2 - no of elements in array 2
Array elements for array2
Output Format
Display the merged array

Sample Input 1

```
5
1
2
3
6
9
4
2
4
5
10
```

Sample Output 1

```
1 2 3 4 5 6 9 10
```

PROGRAM:

```
def merge_arrays_without_duplicates(arr1, arr2):
    # Combine the arrays and convert to a set to remove duplicates
    result_set = set(arr1 + arr2)
    # Convert the set back to a sorted list
```

```
    merged_sorted_array = sorted(result_set)
    return merged_sorted_array

# Input read and processing
def process_input():
    # Reading number of elements and the elements for the first array
    n1 = int(input())
    array1 = []
    for _ in range(n1):
        element = int(input())
        array1.append(element)

    # Reading number of elements and the elements for the second array
    n2 = int(input())
    array2 = []
    for _ in range(n2):
        element = int(input())
        array2.append(element)

    # Merge the arrays without duplicates
    result = merge_arrays_without_duplicates(array1, array2)

    # Print the result
    print(" ".join(map(str, result)))
```

	Input	Expected
✔	5 1 2 3 6 9 4 2 4 5 10	1 2 3 4 5 6 9 10
✔	7 4 7 8 10 12 30 35 9 1 3 4 5 7 8 11 13 22	1 3 4 5 7 8 10 11 12 13 22 3(

Passed all tests! ✔

Ex. No. : 5.9 **Date:**

Register No: 231501049 **Name: GNAANESH B B**

Print Element Location

Write a program to print all the locations at which a particular element (taken as input) is found in a list and also print the total number of times it occurs in the list. The location starts from 1.

For example, if there are 4 elements in the array:
5
6
5
7

If the element to search is 5 then the output will be:
5 is present at location 1
5 is present at location 3
5 is present 2 times in the array.
Sample Test Cases

Test Case 1
Input

4
5
6
5
7
5

Output

5 is present at location 1.
5 is present at location 3.
5 is present 2 times in the array.

Test Case 2
Input
5
67
80

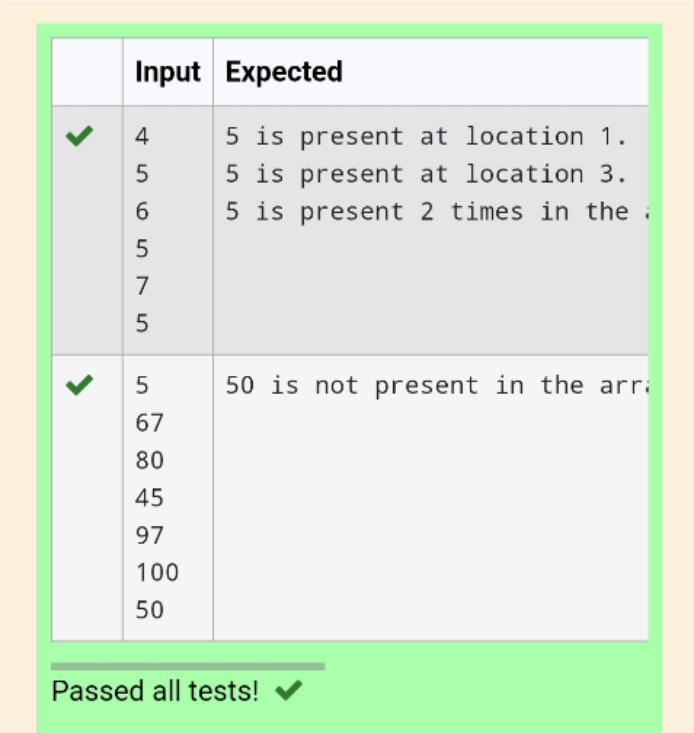
45
97
100
50

Output
50 is not present in the array.

PROGRAM:

```
def find_element_locations(lst, target):
    locations = []
    count = 0
    for i in range(len(lst)):
        if lst[i] == target:
            locations.append(i + 1)
            count += 1

    return locations, count


def main():
    n = int(input())
    lst = [int(input()) for _ in range(n)]
    target = int(input())


    locations, count = find_element_locations(lst, target)


    if count == 0:
        print(f"{target} is not present in the array.")
```

```
    else:
        for loc in locations:
            print(f"{target} is present at location {loc}.")
        print(f"{target} is present {count} times in the array.")

if __name__ == "__main__":
    main()
```



	Input	Expected
✔	4 5 6 5 7 5	5 is present at location 1. 5 is present at location 3. 5 is present 2 times in the a
✔	5 67 80 45 97 100 50	50 is not present in the arra

Passed all tests! ✔