	Dogo of or of the control of the co	S
, 5 3 ^S	STUDENT REPORT	>
CD003	STUDENT REPORT	300
,	Solving State Control of the S	3BRV
DE	ETAILS (536) (500) (500))
(2) I	ETAILS Name SOUNDENT REPORT PAVAN KUMAR M JANGANI	0000
	PAVAN KUMAR M JANGANI	300
00	Roll Number	
3R25	PAVAN KUMAR M JANGANI Roll Number 3BR23CD065	
EX	3BR23CD065 (PERIMENT AND ADDRESS OF COMBINATIONS LEADING TO A PRODUCT COM	200,3
53 Tit	tle 300 340 065 3500 3400 3600 3400 3400 3400 3400 3400 34	
3000	NUMBER OF COMBINATIONS LEADING TO A PRODUCT	873CV
	Completed Complete to At Robbits	36,
S S S S S S S S S S S S S S S S S S S	Description 3 Active Active Active Active Active Active Active Active Active	
6	ABR23CD065 CPERIMENT To be a seription Problem Statement: You are given an array arr and a product m. Your task is to find the number of possible unique triplets whose product of	300005
6	You are given an array arr and a product m. Your task is to find the number of possible unique triplets whose product of	5
3R23CD01	elements is m.	2
\$,	Input Format:	365 3BR
3	 The first line contains the integer, n The second line contains space seperated integers of the array, arr 	,
, choos st	The second line contains space separated integers of the array, arr The third line contains the product m.	CO.
, "	The input will be read from the STDIN by the candidate	BRAGO
23	Output Format:	
Sapra	The output consists of a single integer, i.e. the count of unique triplets having product m.	300005
	The output will be matched to the candidate's output printed on the STDOUT	3000
3R23CD0'	Example:	
3223	Input:	165 3BR
	7	200
CDOPP	5 3 20 10 1 4 2	(
CDC	60	A STORY
0	Output:	Brit
38223	3	0
	Explanation:	Contraction of the second
	Product m:60	,
	Possible triplets for product m: (5,4,3),(20,3,1), (10,3,2)	2
	The count of unique triplets is 3.	Constitution of the consti
	Source Code:	O .
•	Source code.	- F. S.
	Source Code: Shell of the state of the stat	ART AS

```
def count_triplets(arr, n, m):
       unique_triplets = set()
       for i in range(n):
           for j in range(i + 1, n):
               for k in range(j + 1, n):
                    if arr[i] * arr[j] * arr[k] == m:
                       triplet = tuple(sorted([arr[i], arr[j], arr[k]]))
                       unique_triplets.add(triplet)
       return len(unique_triplets)
   # Input Reading
   n = int(input())
   arr = list(map(int, input().split()))
   m = int(input())
   result = count_triplets(arr, n, m)
   print(result)
RESULT
 6 / 6 Test Cases Passed | 100 %
```