	<b>□Logo</b>	٥
123BR235	Exame 12 36 17 36	5.
12-38.	ALTO SERVICE STOPPENT REPORT	38R21
, (	STUDENT REPORT  STUDENT REPORT  STAILS  SAME AND SHAPE AND TO SHAPE SHAP	2172
DE	ETAILS 38 38 AND	_^
SPRI I	Name 123 BELL 123 PRINT BELL 111	D BR23A
	V K SRIJA	
3A1723	3BR23AI172	12
	172 3R 1823 1236 1171	J3A11.
EX	PERIMENT, 3423 ALTO 342 ALTO 3423 AL	P. C.
3º Titl	le 3211 1238 1238 123811	3BR'
,	NUMBER OF COMBINATIONS LEADING TO A PRODUCT	ANTIL
3BR23A1	3BR23Al172  PERIMENT, BER 3AL172 3BR23AL172	3. R. 172. 3. R. 4
3BR	Problem Statement:	38R238
(		ν <sup>'5</sup>
2381172	elements is m.	. 17
	Input Format:	2232177
11723BR	• The first line contains the integer, it	A172385
	The input will be read from the STDIN by the candidate	ALTI
23P	Output Format:	
13BR23P	The output consists of a single integer, i.e. the count of unique triplets having product m.	1238R23
	The output will be matched to the candidate's output printed on the STDOUT	(2)
,R23A117	Example:	^1
,22	Input:	3R23R17
	7	Ø*
,A117238	5 3 20 10 1 4 2	2
, 8		35/8/273
3BR23	Output:	P .
3BF		RAG
	Explanation:  Product m:60	A REAL PROPERTY OF THE PROPERT
		Ā
	Possible triplets for product m: (5,4,3),(20,3,1), (10,3,2)  The count of unique triplets is 3.	38
	The count of unique utplets is a.	3.78
\$	Source Code: 38F2 3F11123 F123F1123F123F1123F123F123F123F123F123F1	RA RELIEF

```
def count_triplets(arr, n, m):
       unique_triplets = sets()
       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)
   n = int(input())
   arr = list(map(int, input().split()))
   m = int(input())
   result = count_triplets(arr, n, m)
   print(result)
RESULT
 0 / 6 Test Cases Passed | 0 %
```