A Tricky Problemm

Sid is bored of getting a lot of homework at tution. So, one day seeing this , his teacher decided to cheer Sid up a bit and gave him a problem to solve. His teacher told him that if he could solve the problem and provide him with the correct answer , Sid can have a day off the next day and he won't be provided with any homework for that day . The problem is as follows :

Sid will be provided with Q integers . On each interger N, he can perform the following operations :

1) Reduce the number by 1 .

2) If N can be represented in the form of N = p*q ,(p!=1, q!=1), if both p and q have same parity , make N = max(N/q ,N/p) else make N = p + q.

Note : By same parity , it is meant that both p and q are either odd or both are even .

Sid needs to find the cube of minimum number of conversion it will take to convert N to 1 .

Now , inorder to solve this problem, Sid need your help . So, write a code inorder to generate the answer and help Sid in getting a day off from his study routine .

Input Format

The first line contains an integer Q , denoting number of queries . The next Q lines contains integer N .

Constraints

1<=Q<=10^3

1<=N<=10^7

Output Format

Print the cube of minimum numbers of steps required to be performed on N to reach 1 .

Sample Input 0

```
2
5
10
```

Sample Output 0

```
27
64
```

Explanation 0

You can convert 5 to 1 in minimum 3 steps . 5->4->2->1 . So, answer will be 3^3^3 i.e 27

You can convert 10 to 1 in minimum 4 steps . 10->9->3->2->1 . So, answer will be 4*4*4 i.e 64 . As it can be seen, greedy approach will lead to a wrong answer in this case . Using greedy approach we would get conversion as 10->7->6->5->4->2->1 .

Current Buffer (saved locally, editable) C++

```cpp
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;


int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}
```

Line: 1 Col: 1