Periodic function, trip 1

xkcd/26

Let us consider periodic functions from **Z** to **R**.

```
def f(x): return [4, -6, 7][x%3] # (with Python notations)
# 4, -6, 7, 4, -6, 7, 4, -6, 7, 4, -6, 7, 4, -6, 7, ...
```

For example, f is a 3-periodic function, with $f(0) = f(3) = f(6) = f(9) = 4$.
With a simplified notation we will denote f as [4, -6, 7].

For two periodic functions (with integral period), here the quotient of periods will be rational, in that case it can be shown that the sum of the functions is also a periodic function.
Thus, the set of all such functions is a vector space over **R**.

Our interest, in this problem, will be the dimension of this space when the period is bounded by some integer N.

Input

The first line contains an integer T, the number of test cases.
On the next T lines, you will be given an integer N.
Consider the family of all n-periodic functions for n in [1..N]. There are some links between some functions.
For example: [2, 0] = [2, 0, 1, 0] + [0, 0, 1, 0], with simplified notations.

Output

Print the rank of this family ; ie the size of the largest collection of **R**-linearly independent elements of this family.

Example

Input:
3
2
3
4

Output:
2
4
6

Constraints

0 < T < 10^2
0 < N < 10^8

There's two input files, one easy (mostly small input), and a hard one (uniform random input).

My PY3.4 code get AC in 0.03+0.89=0.92s. This code isn't optimized.
I suspect there are several competitive approaches for this task.
Have fun ;-)