While Steven is teaching Minimum Spanning Tree in CS3233, Mr. K proudly claimed that he can find the Minimum Spanning Tree of a graph with vertices up to 1 million. Steven, doubtful of the claim, asked Mr. K if it works for dense graphs as well. “Of course!” Mr. K exclaimed without thinking.

Now Mr. K needs to demonstrate his algorithm, and realizes he cannot. He quickly clarifies, “What I meant was that I can find the Minimum Spanning Tree of a dense graph with some conditions…” Specifically, the graph Mr. K is talking about fulfills the following conditions:

- There are **N-1** vertices in the graph, labeled {2,3,4,…,N}
- There exists an edge of weight **i+j** between vertices i and j if both i and j are prime
- There exists an edge of weight **GCD(i,j)** between vertices i and j if both i and j are composite and if **GCD(i, j) > 1**
- For all vertices that are not adjacent to anything else, they will have an edge of **weight 7** (Steven’s favorite number) to vertex 2
- If N >= 4, there exists an edge of **weight 7** between vertices 2 and 4

“Alright, I understand your special graph. Now how do you find the MST?” Steven asked. Stumped yet again, Mr. K quickly got the guest trainees from the CS3233 course, known for their abilities to beat NUS undergraduates, to help.

TASK

Help Mr. K find the MST of his special graph given the integer N based on the following constraints. Note that in order to score credits for any of the subtasks, **you have to solve all test cases** that constitute the subtask.

Subtask 1 [10 points]

2 <= N <= 50

Subtask 2 [10 points]

2 <= N <= 100

Subtask 3 [10 points]

2 <= N <= 1,000

Subtask 4 [35 points]

2 <= N <= 1,000,000

Subtask 5 [35 points]

2 <= N <= 10,000,000

FEEDBACK

You have **10** tokens that you can use to release the **complete** feedback to the question.

INPUT

There will be only one integer in the input, denoting N.

OUTPUT

Output the weight of the minimum spanning tree on the first line of the output. **Note that you will need a 64-bit integer to represent your output.**

EXAMPLES

Sample Input	Sample Output
2	0
6	21