문제

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **시간 제한** | **메모리 제한** | **제출** | **정답** | **맞은 사람** | **정답 비율** |
| 1 초 | 64 MB | 519 | 287 | 239 | 62.078% |

## 문제

서강대학교 컴퓨터공학과 실습실 R912호에는 현재 N개의 창문이 있고 또 N명의 사람이 있다. 1번째 사람은 1의 배수 번째 창문을 열려 있으면 닫고 닫혀 있으면 연다. 2번째 사람은 2의 배수 번째 창문을 열려 있으면 닫고 닫혀 있으면 연다. 이러한 행동을 N번째 사람까지 진행한 후 열려 있는 창문의 개수를 구하라. 단, 처음에 모든 창문은 닫혀 있다.

예를 들어 현재 3개의 창문이 있고 3명의 사람이 있을 때,

1. 1번째 사람은 1의 배수인 1,2,3번 창문을 연다. (1, 1, 1)
2. 2번째 사람은 2의 배수인 2번 창문을 닫는다. (1, 0, 1)
3. 3번째 사람은 3의 배수는 3번 창문을 닫는다. (1, 0, 0)

결과적으로 마지막에 열려 있는 창문의 개수는 1개 이다.

## 입력

첫 번째 줄에는 창문의 개수와 사람의 수 N(**1 ≤ N ≤ 2,100,000,000**)이 주어진다.

## 출력

마지막에 열려 있는 창문의 개수를 출력한다.

## 예제 입력 복사

3

## 예제 출력 복사

1

## 예제 입력 2 복사

24

## 예제 출력 2 복사

4

## 힌트

## 출처

제출 1 - 런타임 에러 (배열길이초과)

package alogprithm;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.OutputStreamWriter;

import java.util.ArrayList;

public class T\_13909 {

public static void main(String[] args) throws NumberFormatException, IOException {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

// BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(System.out));

int people = Integer.parseInt(br.readLine());

ArrayList<Boolean> window = new ArrayList<>();

int count = 0;

for(int i=0;i<people;i++) {

window.add(true);

count++;

}

for(int i=1;i<window.size();i++) {

for(int j=i+1;j<=window.size();j\*=(i+1)) {

if(window.get(j-1)) {

window.set(j-1, false);

count--;

}else {

window.set(j-1, true);

count++;

}

}

// System.out.println(window);

}

System.out.println(count);

// bw.write(count);

// bw.flush();

}

}

제출 2 - 시간 초과

package alogprithm;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class T\_13909\_2 {

public static void main(String[] args) throws NumberFormatException, IOException {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int people = Integer.parseInt(br.readLine());

int count = 1;

int check = 0;

for (int i = 2; i <= people; i++) {

for (int j = 2; j <= (i / 2); j++) {

if (i % j == 0) {

if (j \* j == i) {

check++;

} else {

check += 2;

}

}

}

if (check % 2 == 1) {

count++;

}

check = 0;

}

System.out.println(count);

}

}

제출 3 -- 런타임 반으로 줄였으나 그래도 실패

package alogprithm;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class T\_13909\_2 {

public static void main(String[] args) throws NumberFormatException, IOException {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int people = Integer.parseInt(br.readLine());

int count = 1;

int check = 0;

for (int i = 2; i <= people; i++) {

for (int j = 2; j <= (i / 2); j++) {

if (i % j == 0) {

if (j \* j == i) {

check++;

} else {

check += 2;

}

}

}

if (check % 2 == 1) {

count++;

}

check = 0;

}

System.out.println(count);

}

}

제출 4 - good

package alogprithm;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class T\_13909\_3 {

public static void main(String[] args) throws NumberFormatException, IOException {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int people = Integer.parseInt(br.readLine());

int count = 0;

int check = 3;

while(true) {

count ++;

people -= check;

check+=2;

if(people <= 0) {

break;

}

}

System.out.println(count);

}

}