import java.io.\*;

import java.util.\*;

import java.text.\*;

import java.math.\*;

import java.util.regex.\*;

import java.math.BigInteger;

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;

//---- Do not change the code.....//

class Counter{

int count = 0;

long threadId = 999999;

void add(){

if(threadId != Thread.currentThread().getId()){

threadId = Thread.currentThread().getId();

}else{

System.out.println("Error : Same thread id cannot proceed");

return;

}

for (int i = 0; i < 100000; i++) {

count++;

}

try{

Thread.sleep(1);

}

catch(Exception e){

}

}

int getTotalCount(){

return count;

}

}

class ThreadRun extends Thread

{

Counter counter = new Counter();

public ThreadRun(Counter counter)

{

this.counter = counter;

}

public void run()

{

counter.add();

}

}

public class Solution {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int intiger = scan.nextInt();

Counter counter = new Counter();

int thread\_count = intiger / 100000;

ThreadRun[] new\_thread = new ThreadRun[thread\_count];

try{

for(int thread = 0; thread < thread\_count; thread++)

{

new\_thread[thread] = new ThreadRun(counter);

new\_thread[thread].start();

Thread.sleep(2);

}

}

catch(Exception e){

}

try{

for(int thread = 0; thread < thread\_count; thread++)

new\_thread[thread].join();

}

catch(Exception e){

}

// Solution is the main Class. Please write your code here, you can also create new classes.

//Hint : 1. Use counter object in line no 46 and span threads that will call the counter.add() method internally.

// 2. Wait for all threads to finish.

// learn to create Threads in Java : https://www.javatpoint.com/creating-thread

//dont not change this code; this is for output

System.out.println(counter.getTotalCount());

}

}