import java.io.\*;

import java.lang.\*;

import java.util.concurrent.locks.\*;

public class Bank{

private int balance=1000;

public static void main(String args[]) throws Exception{

BufferedReader console = new BufferedReader(newI nputStreamReader(System.in));

Bank bank= new Bank();

bank.balance = Integer.parseInt(console.readLine());

DepositThread deposit=new DepositThread(bank);

WithdrawThread withdraw = new WithdrawThread(bank);

deposit.start();

withdraw.start();

deposit.join();

withdraw.join();

System.out.println("all done: balance = "+bank.balance);

}

private static ReentrantLock lock = new ReentrantLock();

public void deposit(){

try{

lock.lock();

for(int i=0;i<1e7;i++){

balance++;

}

}finally{

lock.unlock();

}

}

public void withdraw(){

try{

lock.lock();

for(int i=0;i<1e7;i++){

balance--;

}

}finally{

lock.unlock();

}

}}

class DepositThread extends Thread{

Bank bank;

DepositThread(Bank bank){

this.bank = bank;

}

public void run(){

bank.deposit();

}

}

class WithdrawThread extends Thread{

Bank bank;

WithdrawThread(Bank bank){

this.bank = bank;

}

public void run(){

bank.withdraw();

}

}