

## Problem Statement

You have a deck of cards containing **n** red and **m** black cards. Initially, you have **c** chips. The cards are shuffled and then dealt one after another. Before each card is dealt, you place a bet of **B** chips (where **B** is between 0 and the number of chips you have left, inclusive) and predict whether that card will be red or black. If your prediction is correct, you receive back the **B** chips that you bet and win an additional **B** chips. If your prediction is incorrect, you lose the **B** chips that you bet.

You are a cautious player, so your strategy is to play in such a way that maximizes the number of chips you have at the end of the game in the worst possible case. In other words, your strategy must guarantee that you end the game with at least **X** chips, regardless of the order of cards in the deck, where **X** is as large as possible. Return this value of **X**.

For example, let **n** = 1, **m** = 2 and **c** = 3. The following optimal strategy will guarantee that you end the game with at least 8 chips: Bet 1 chip that the first card is black. If it is, you will then have 4 chips. Skip the next card, and bet all 4 chips on the last card. You will know the color of that last card because you will have already seen the other two cards. If you lose your first bet, you will have 2 chips left, and the two remaining cards will be black. Bet all your chips on each of the remaining cards, and you will double your chips twice, for a total of 8 chips. No other strategy can guarantee more than 8 chips in the worst case.

## Constraints

- **n** and **m** will each be between 0 and 10, inclusive.
- **c** will be between 1 and 30, inclusive.

**Input: n m c**

**Output: X**

## Example

input	output	explain
1 2 3	8	From problem statement
10 0 30	30720	This time all cards are red, so you must bet everything each time.
1 1 1	2	Do not bet on the first card (or you can lose everything).