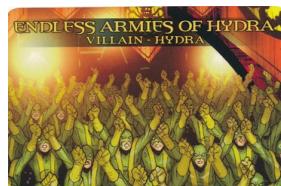


## B. Bydra

time limit per test: 1 second

memory limit per test: 256 megabytes

Na Hydra há o seguinte lema: "Corta-se uma cabeça, duas surgirão em seu lugar".



Cansado de esperar pelo Capitão América invadir as instalações da Hydra, Armin Yolo resolveu jogar o seu jogo preferido para passar o tempo: Bydra, o baralho da Hydra.

O jogo funciona da seguinte forma: nesse baralho toda carta possui um valor inicial e um valor atual, que inicialmente são iguais; a carta que encontra-se no topo do baralho é selecionada; essa carta selecionada é descartada e, a partir do seu descarte, dois cenários podem acontecer:

1. Duas cartas iguais, subtraídas de uma unidade do seu valor atual são colocadas no topo do baralho, exceto se o seu valor atual estiver indo a zero após a subtração; ou
2. Se seu valor atual chega a zero, apenas o seu valor inicial é registrado na mesa, e nenhuma carta é adicionada no topo do baralho.

Como o jogo pode demorar bastante, Armin pediu a sua ajuda para escrever um programa que, dada a disposição inicial das cartas de bydra, indicar qual será a disposição final registrada das cartas à mesa.

### Input

A entrada contém um único caso de teste. A primeira linha contém um inteiro  $N$  ( $1 \leq N \leq 54$ ) que indica o número de cartas no início da rodada. A segunda linha contém  $N$  inteiros  $c_i$  ( $1 \leq c_i \leq 10$ ), separados por um espaço, que indicam o valor inicial de cada carta no baralho.

### Output

A saída deverá conter múltiplas linhas, onde cada linha deverá conter registrado o valor inicial da carta que atingiu valor zero.

### Examples

<b>input</b>	<input type="button" value="Copy"/>
2 1 3	
<b>output</b>	<input type="button" value="Copy"/>
3 3 3 3 1	

<b>input</b>	<input type="button" value="Copy"/>
4 2 1 1 4	
<b>output</b>	<input type="button" value="Copy"/>
4 4	

### IDP - TAA - 2025/02

Private

Participant



### → About Group



Este grupo tem o objetivo de organizar as atividades de programação da disciplina de Técnicas de Programação e Análise de Algoritmos.

[Group website](#)

### → Group Contests

- TAA - LEA 05
- TAA - LEE 05
- TAA - LEA 04
- TAA - LEE 04
- TAA - AS 01
- TAA - LEA 03
- TAA - LEE 03
- TAA - LEA 02
- TAA - LEE 02
- TAA - LEA 01
- TAA - LEE 01
- ET - Exercício de Testes

### TAA - AS 01

Finished

Practice



### → Submit?

Language: [GNU G++17 7.3.0](#)

4  
4  
4  
4  
4  
4  
4  
1  
1  
2  
2

Choose file:

Escolher Arquivo

Nenh...colhido

Submit

### Note

No primeiro caso de testes, há duas cartas no baralho, onde a leitura do topo do baralho para o fundo do mesmo são 3 e 1 (na organização da entrada do problema, lê-se do fundo para o topo como 1 3). Inicialmente, retira-se a carta do topo do baralho (3), como o seu valor menos uma unidade não é zero, adiciona-se duas cartas subtraídas de uma unidade da carta descartada. A configuração atual do baralho passa a ser a seguinte: 1 2 2. Em seguida, descarta-se a carta do topo e realiza-se a mesma análise, como o seu valor menos uma unidade não é zero, adiciona-se duas cartas subtraídas de uma unidade da carta descartada, ficando o seguinte: 1 2 1 1. No próximo passo, a carta descartada (1) subtraída de uma unidade tem seu valor igual a zero, então registra-se na mesa o seu valor inicial que é 3. Resta agora no baralho 1 2 1, o qual continuam-se os passos até que o mesmo encontre-se vazio.

---

[Codeforces](#) (c) Copyright 2010-2025 Mike Mirzayanov

The only programming contests Web 2.0 platform

Server time: Nov/11/2025 18:28:35<sup>UTC-3</sup> (j2).

Desktop version, switch to [mobile version](#).

[Privacy Policy](#) | [Terms and Conditions](#)

Supported by



ITMO