



## शेफ एंड मैजिकल जार्स (Chef and Magical Jars)

Problem code: MAGICJAR

शेफ ने जूनियर शेफ को कुछ एडवांस रेसिपी सिखाने का फैसला किया है। सेशन के पहले दिन, प्रत्येक जूनियर शेफ की खाना पकाने की क्षमता की पहचान करने के लिए, शेफ ने उन्हें अपना प्रमुख नुस्खा बनाने का निर्देश दिया। जूनियर शेफ बहुत उत्साहित थे और वे सभी अपने प्रमुख नुस्खा को पकाने के लिए रसोई में चले गए।

शेफ के पास रसोई में सीमित संख्या में जार हैं। जार जादुई होते हैं - अगर कोई शेफ पकवान पका रहा हो जिसके लिए  $x$  सामग्री आवश्यक है, तो वह  $x$  जार लेता है, इनमें से प्रत्येक जार अपने आप को आवश्यक सामग्री से भर देता है, और इसके बाद शेफ खाना पकाने का काम पूरा करता है और जार को रसोई में लौटाता है, वे खुद को खाली करते हैं। (और अन्य शेफ के लिए कुछ या सभी लेने के लिए तैयार हैं)। बेशक, इसका मतलब यह है कि  $x$  जार से कम  $x$  सामग्री की आवश्यकता वाले डिश को पकाना असंभव है।

चूँकि शेफ ने किसी भी क्रम के जूनियर शेफ को निर्देश नहीं दिया था कि वे जार उठाएँ, इसलिए उन्होंने एक साथ जार चुनना शुरू किया और ऐसा हुआ कि जूनियर शेफ में से कोई भी अपना नुस्खा तैयार करने के लिए पर्याप्त जार प्राप्त नहीं कर पाया। साथ ही उनमें से कोई भी अपना नुस्खा पूरा करने के लिए अपने जार को दूसरे जूनियर शेफ को देने के लिए तैयार नहीं था। शेफ इस स्थिति को संभालने में सक्षम नहीं थे और उस दिन के लिए सेशन बंद करने का फैसला किया, ताकि वह अधिक जार प्राप्त कर सकें और बाद में फिर से प्रयास करें।

आप जानते हैं कि एन जूनियर शेफ (1 से  $N$  नंबर) हैं और प्रत्येक वैध  $i$  के लिए,  $i$ -th शेफ के पकवान के लिए आवश्यक सामग्री की संख्या  $A_i$  है। यदि  $J$  जार हैं, तो औपचारिक रूप से, निम्नलिखित प्रक्रिया होती है:

- जूनियर शेफ कुछ जार लेते हैं; आइये  $a_i$  द्वारा  $i$ -th शेफ द्वारा लिए गए जार की संख्या को निरूपित करते हैं। जार का कोई वितरण जैसे कि  $0 \leq a_i$  प्रत्येक वैध  $i$  के लिए और 
$$\sum_{i=1}^N a_i = J$$
 संभव है।
- किसी भी समय, यदि  $a_i < A_i$  प्रत्येक शेफ  $i$  के लिए जो कि अभी तक उनकी डिश तैयार नहीं की है, तो खाना पकाने का सेशन एक विफलता है।

- अन्यथा, शेफ में से एक जिनके पास आवश्यक सामग्री की संख्या के रूप में कम से कम उतने जार हैं, वे अपने पकवान तैयार करते हैं और अपने जार को रसोई में वापस कर देते हैं।
- जब भी कुछ जार रसोई में वापस आते हैं, तो उन्हें तुरंत कुछ शेफ द्वारा ले जाया जाता है, जिन्होंने अभी तक अपने व्यंजन तैयार नहीं किए हैं (संभवतः सभी जार एक शेफ द्वारा)।
- यह प्रक्रिया शेफ के जार लेने, उनके व्यंजन पकाने और जार लौटने तक जारी रहती है, जब तक कि कोई भी शेफ अपने पकवान को नहीं पका सकता है या सभी रसोइयों ने अपने व्यंजन नहीं बनाए हैं।
- जब सभी जूनियर शेफ ने अपने व्यंजनों को सफलतापूर्वक पकाया है, तो खाना पकाने का सेशन सफलतापूर्वक समाप्त होता है।

शेफ चाहते हैं कि कम से कम जादुई जार पता चले जो शुरुआत में रसोई में मौजूद होना चाहिए ताकि जूनियर शेफ जारों को कैसे भी उठाएं, जैसे की ये इस सेशन सफल हों। शेफ एक लीजेंडरी कुक है, लेकिन वह गणित में बहुत अच्छा नहीं है, इसलिए वह आपसे उस नंबर को खोजने के लिए कहता है।

### इनपुट:

- इनपुट की पहली लाइन में एक सिंगल इन्टिजर  $T$  है जो टेस्ट केसेस की संख्या को दर्शाता है।  $T$  टेस्ट केसेस का विवरण निम्नानुसार है।
- प्रत्येक टेस्ट केस की पहली लाइन में एक सिंगल इन्टिजर  $N$  है।
- दूसरी लाइन में स्पेस के साथ  $N$  इन्टिजर है -  $A_1, A_2, \dots, A_N$  |

### आउटपुट:

प्रत्येक टेस्ट केस के लिए, एक इन्टिजर युक्त एक लाइन को प्रिंट करें - जार की न्यूनतम संख्या ।

### बाध्यता\Constraints :

- $1 \leq T \leq 1,000$
- $1 \leq N \leq 10^5$
- $1 \leq A_i \leq 10^9$  प्रत्येक मान्य  $i$  के लिए
- सभी टेस्ट केसेस में  $N$  का सम  $10^6$  से अधिक नहीं है

## सब्टास्क

सब्टास्क #1 (100 points): ओरिजिनल बाध्यताएँ

### इनपुट उदाहरण

2

4

1 1 1 1

2

1 4

### आउटपुट उदाहरण

1

4

### उदाहरण का स्पस्टीकरण

**उदाहरण केस #1:** कोई भी एक जूनियर शेफ 1 जार उठा सकता है और नुस्खा पका कर और इसे वापस रसोई में रख सकता है। ताकि, अन्य जूनियर शेफ उसी जार का पुनः उपयोग कर सकें और नुस्खा पका सकें। इस प्रक्रिया को प्रत्येक जूनियर शेफ द्वारा दोहराया जा सकता है ताकि वे अपने नुस्खा को पका सकें।