[Unique Sandwich]

[Houdaifa Bouamine]

Problem Statement

A famous restaurant, **Sefar**, is hosting multiple feasts this year. For each feast, the restaurant will serve **unique triangular sandwiches** to its guests. Each sandwich is defined by three integer side lengths **a,b,c** must form a valid triangle (flat triangles are not valid) and unique within the same feast, and **N** = **a** + **b** + **c** is the **perimeter** of the sandwich (length of bread the cheese sandwich it) must be equal for all guests on a single feast.

To ensure all **k** guests at a feast receive unique triangular sandwiches, you need to calculate the **minimum perimeter N** such that there are at least k unique sandwiches with perimeters equal to N.

notice that the triples (3,3,5) and (3,5,3) are considered the same

Input:

- The first line contains **T**, the number of test cases.
- For each test case:
  - The first line contains **m**, the number of feasts held that year.
  - The next **m** lines each contain **k1, k2, k3, …km,** where **ki** is the number of guests for the **i-th** feast.

Output:

- For each feast in each test case, find the single integer **N** representing the minimum perimeter required for the given number of customers, and generate the flag from it.

Examples :

Example 1 :

Input:

```
2
3
10 18 32
4
89 10 3 32
```

Results (each line is the result of each test case):

```
19 27 37
63 19 9 37
```

flag:
5539e8b27f3becc7e68fc66a928e7d6ff0ef8b92f1de066a7b16eba31852b5a4

Example 2 :

Input:

```
3
5
95 73 27 31 83
10
63 150 78 91 165 99 95 455 117 296
12
194 9495 5207 9036 8944 7995 5055 3763 6213 4994 396 7989
```

Results (each line is the result of each test case):

```
65 57 33 37 61
53 83 59 63 87 67 65 145 73 117
95 673 497 657 653 617 491 423 545 487 135 617
```

flag:
227b4335921e5f9618ff992918a4cf42658d8022598a3efe8060c523ab6b7266

How to get the flag?:

From the 1st example:

First concatenate the all the results (example from prev testcase) :

```
19 + 27 + 37 + 63 + 19 + 9 + 37 => 1927376319937
```

Then pass it to SHA256 function:

sha256(1927376319937) =>
5539e8b27f3becc7e68fc66a928e7d6ff0ef8b92f1de066a7b16eba31852b5a4

Expected performance:

a solution for **each feast** with complexity **O(log(n))** will pass in a few **seconds**, **O(n)** will take a few **minutes**, slower solutions may not pass with in the competition duration.