

June Training Dynamic Programming Contest

Problem 1: *examhalls*

Problem Description

RI is currently having its Mid-year Common Tests. Students sitting for the exam papers are required to sit for it in one of two halls: either the Shaw Foundation Hall, or the Albert Hong Hall. There are N classes currently required to sit for the exams, and the students of each class have to sit in the same exam hall.

For the exam, the school administration ranks the difficulty of administering a paper in an exam hall to be the square of the number of students in the exam hall. In other words if A = the number of students in Albert Hong Hall and B = the number of students in Shaw Foundation Hall, the difficulty of administering the paper will be $A^2 + B^2$ in total.

Take a situation where $N = 5$, and the set of the class sizes = $\{3, 4, 1, 2, 5\}$.

The best way to split the students is to split them up as follows: $\{1, 2, 4\}$ into Albert Hong Hall, and $\{3, 5\}$ into Shaw Foundation Hall. This will result in a minimal difficulty of $(1+2+4)^2 + (3+5)^2 = 113$. It can be proven that there are no better solutions that can achieve a difficulty of less than 113.

Given N , and the number of students in each of the N classes, calculate the minimum difficulty of administering the paper possible.

Implementation

Your program is required to include the header file `examhalls.h` as part of your code.

In addition, your program will be required to implement the functions:

- `long long examhalls(int N, int A[])`, which should return the minimum difficulty of the paper where N refers to the number of classes and A contains an array of the class size for each class.

Please **do not** input or output anything to or from standard input (stdin) or standard output (stdout). Submissions that do so will be graded as **Incorrect**.

Grading

All submissions should adhere to the following time and memory constraints:

- Time Limit: 1 second.
- Memory Limit: 64 Megabytes.

All testdata for this problem will satisfy the following conditions:

- $1 \leq N \leq 1\,000$.
- The total number of students taking the exam will not exceed 50 000.

In addition, partial credit can also be obtained by passing **all** testdata from the subtasks:

Subtask 1 (13 points)

- $1 \leq N \leq 20$.
- The total number of students taking the exam will not exceed 1 000.

Subtask 2 (21 points)

- $1 \leq N \leq 100$.
- The total number of students taking the exam will not exceed 1 000.

Subtask 3 (27 points)

- $1 \leq N \leq 1\,000$.
- The total number of students taking the exam will not exceed 1 000.

Subtask 4 (39 points)

- $1 \leq N \leq 1\,000$.
- The total number of students taking the exam will not exceed 50 000.

Testing

In order to help you in testing your submission, *grader.cpp* will be provided. A starting template *examhalls.cpp* will also be provided. It should be compiled together with your submission and run with the sample inputs to test your program before submission.

Sample Input 1

```
5
3 4 1 2 5
```

Sample Output 1

```
113
```

Sample Input 2

```
7
7 11 2 3 4 9 2
```

Sample Output 2

```
722
```