

## Feedback — Programming Question - 6

[Help Center](#)

You submitted this homework on **Fri 27 Feb 2015 3:06 PM IST**. You got a score of **5.00** out of **5.00**.

### Question 1

Download the text file [here](#). (Right click and save link as).

The goal of this problem is to implement a variant of the 2-SUM algorithm (covered in the Week 6 lecture on hash table applications).


The file contains 1 million integers, both positive and negative (there might be some repetitions!). This is your array of integers, with the  $i^{th}$  row of the file specifying the  $i^{th}$  entry of the array.

Your task is to compute the number of target values  $t$  in the interval  $[-10000, 10000]$  (inclusive) such that there are *distinct* numbers  $x, y$  in the input file that satisfy  $x + y = t$  (NOTE: ensuring distinctness requires a one-line addition to the algorithm from lecture.)

Write your numeric answer (an integer between 0 and 20001) in the space provided.

OPTIONAL CHALLENGE: If this problem is too easy for you, try implementing your own hash table for it. For example, you could compare performance under the chaining and open addressing approaches to resolving collisions.

You entered:

Your Answer	Score	Explanation
427	 2.00	

Total

2.00 / 2.00

## Question 2

Download the text file [here](#).

The goal of this problem is to implement the "Median Maintenance" algorithm (covered in the Week 5 lecture on heap applications). The text file contains a list of the integers from 1 to 10000 in unsorted order; you should treat this as a stream of numbers, arriving one by one. Letting  $x_i$  denote the  $i$ th number of the file, the  $k$ th median  $m_k$  is defined as the median of the numbers  $x_1, \dots, x_k$ . (So, if  $k$  is odd, then  $m_k$  is  $((k+1)/2)$ th smallest number among  $x_1, \dots, x_k$ ; if  $k$  is even, then  $m_k$  is the  $(k/2)$ th smallest number among  $x_1, \dots, x_k$ .)

In the box below you should type the sum of these 10000 medians, modulo 10000 (i.e., only the last 4 digits). That is, you should compute  $(m_1 + m_2 + m_3 + \dots + m_{10000}) \bmod 10000$

OPTIONAL EXERCISE: Compare the performance achieved by heap-based and search-tree-based implementations of the algorithm.

**You entered:**

Your Answer		Score	Explanation
1213	✓	3.00	
Total		3.00 / 3.00	

