

Problem 4: The median

is the middle value in an ordered integer list. If the size of the list is even, there is no middle value, and the median is the mean of the two middle values.

```
import heapq
class MedianFinder:

    def __init__(self):

        self.max_heap = []

        self.min_heap = []

    def addNum(self, num: int) -> None:

        heapq.heappush(self.max_heap, -num)

        heapq.heappush(self.min_heap, -heapq.heappop(self.max_heap))

        if len(self.min_heap) > len(self.max_heap):

            heapq.heappush(self.max_heap, -heapq.heappop(self.min_heap))

    def findMedian(self) -> float:

        if len(self.max_heap) == len(self.min_heap):

            return (-self.max_heap[0] + self.min_heap[0]) / 2

        else:

            return -self.max_heap[0]

medianFinder = MedianFinder()

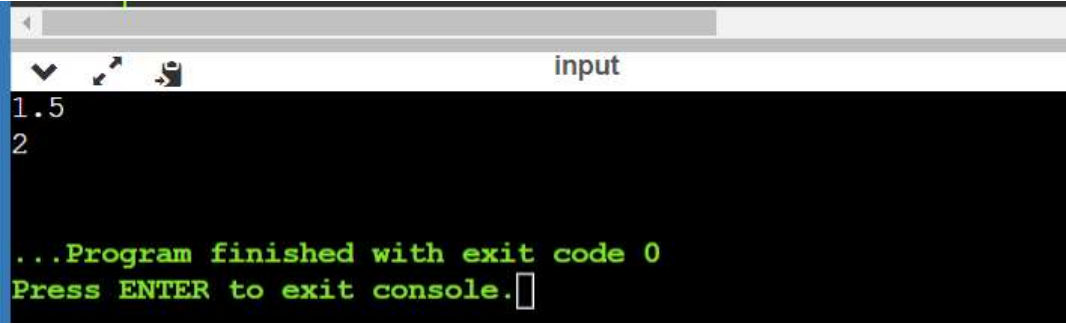
medianFinder.addNum(1)

medianFinder.addNum(2)

print(medianFinder.findMedian())

medianFinder.addNum(3)

print(medianFinder.findMedian())
```



```
input
1.5
2
...Program finished with exit code 0
Press ENTER to exit console.
```