Lab3 Sum of K

We are given input file in.txt that has number K and sequence of N numbers. We want to determine if there are two numbers whose sum equals a given number K. For instance, if the input file is

SumOfK

10

8 4 1 6

We know from the file that K is 10, sequence of numbers is 8 4 1 6, and number of elements N is 4 (we can count numbers)

For given information then the answer is yes (4 and 6), because 4+6 is 10.

A number may be used twice.

If input file is

SunOfK

10

8 4 5 3

the answer is also yes, because 5+5 is 10

Do the following:

Algorithm 1 : Give an $O(N^2)$ algorithm to solve this problem.

Algorithm 2: Give an O(N*log(N)) algorithm to solve this problem. (Hint: Sort the items first. After that is done, you can solve the problem in linear time.)

https://en.wikipedia.org/wiki/Heapsort

Code both solutions and compare the running times of your algorithms.

Get input from input file 1_in.txt

Print output to output file 1_out.txt

For input 1_in.txt

```
SumOfK
13
5 12 8 10 7 4 3 5 5 3 2 1
```

Output file 1_out.txt has to have:

```
13
Before heap sort
```

```
SumOfK
5 12 8 10 7 4 3 5 5 3 2 1
Algorithm1 calculation only O(n^2)
Yes
5+8=13
Algorithm2 includes build heap and heap sort O(nlog(n))
Calculation itself in Algorithm2 O(n)
Yes
1+12=13
Algortihm1: 7.33e-07 nanoseconds,
Algortihm2: 2.199e-06 nanoseconds,
```

//calculation time in nanoseconds for both algorithms

For input 2_in.txt

```
SumOfK
17
5 12 17 10 7 4 3 5 5 3 2 1
```

Output file has to have:

```
17
Before heap sort
SumOfK
5 12 17 10 7 4 3 5 5 3 2 1
Algorithm1 calculation only O(n^2)
Yes
5+12=17
Algorithm2 includes build heap and heap sort O(nlog(n))
Calculation itself in Algorithm2 O(n)
Yes
5+12=17
Algortihm1: 0 nanoseconds,
Algortihm2: 2.199e-06 nanoseconds,
```

//calculation time in nanoseconds for both algorithms

For input 3_in.txt

```
SumOfK
200
5 12 8 10 7 4 3 5 5 3 2 1
```

Output file 3_out.txt has to have:

```
200
Before heap sort
SumOfK
5 12 8 10 7 4 3 5 5 3 2 1
Algorithm1 calculation only O(n^2)
No
Algorithm2 includes build heap and heap sort O(nlog(n))
Calculation itself in Algorithm2 O(n)
No
```

```
Algortihm1: 1.466e-06 nanoseconds,
Algortihm2: 2.932e-06 nanoseconds,
```
//calculation time in nanoseconds for both algorithms

Please find input and output files in this folder.

Repeat calculations for 4 input files, named 1_in.txt, 2_in.txt, 3_in.txt, 4_in.txt,

Assume that file 4_in.txt does not exist in your folder. When you try to open the file, and if you did not succeed, you must throw exception, and stop calculations.

Output files are 1_out.txt, 2_out.txt, 3_out.txt correspondently.

Pseudocode for Driver program is in this folder.