Contest	[Fall17][B1] Assignment #2
Problem	HashTable Impementation
Input file(s)	`input.txt`
Output file	`output.txt`
Time limit	10 sec.
Memory limit	128 MB

Statement

Part A: `HashTable Class:`Implement a class HashTable with the following public methods with expected complexity: `get-O(1), put-O(1), remove-O(1), size-O(1), isEmpty-O(1), entrySet-O(n), keyset-O(n), values-O(n).`

Also implement a method called `getMaxProbabingSequenceLength()`, which will return the length of the longest probing sequence.

`Collision Handling:`Hashtable is implemented using open addressing with quadratic probing technique.

Part B: You are given a text file `input.txt` that consists of multiple lines in English. You need to process the entire file to count distinct words (case insensitive) that appear more than once considering the following constraints:

You should use your own `hashtable` implementation from part (A).

You should consider all the words except that are in the following list: {a, in, at, to, on, not, for, s, 's, 'd, 're, is, are, am, has, I, we, you}.

As said earlier, you must use your hashtable to do this task, but at the end of the task your hash table must be empty.

Each line in your result in `output.txt`should consist of the word and its count, as shown in below example. The order of words in `output.txt` is an alphabetical. There's always at least one such word in the input file.

`Note:` The word “Key” and “Keys” are considered as same, and counted as <”key”, 2>.

Example

`input.txt`

```
The ones found cheating in the assignment


Their assignment grade will be set to the zero.
```

`output.txt`

```
<”assignment”, 2>


<”the”, 3>
```

Contest	[Fall17][B1] Assignment #2
Problem	Sorting Techniques
Input file(s)	`input.txt`
Output file	`output.txt`
Time limit	10 sec.
Memory limit	128 MB

Statement
Implement your own `Selection, Quick, Bubble, Insertion, and Merge Sort Class` individually `(each class should be with its name, e.g. Selection, Bubble, etc.)` to Sort an array that contains Numbers and Characters in randomly permuted form. In `your implementation`, you need to consider the index of both integers and characters individually and sort them as per their indexes (See the example).

The `input.txt` contains the unsorted arrays which include both numbers and characters

(alphabets). Your `output.txt` should contain the sorted arrays.

`Note` Sort the arrays from the randomly permuted form and Do not use concatenation after sorting, because it will discard their indexes.

Example

`input.txt`

```
10 15 A 2 D L 20 1 5 C 9 3 Z R N 14
```

`output.txt`

```
1 2 A 3 C D 5 9 10 L 14 15 N R Z 20
```

Contest	[Fall17][B1] Assignment #2
Problem	[STRATEGY] White Rabbit
Input file(s)	`input.txt`
Output file	`output.txt`
Time limit	10 sec.
Memory limit	128 MB

Statement

White rabbit. needs to buy some time, so he rushed into a clock shop. But he has only a few cents. Help him to estimate how many seconds can he buy in the best case, if you know what

time clocks have (in seconds), their price (in cents) and Rabbit's budget (in cents).

The first line of `input.txt` stores repeating pairs of type time1 cost1, time2 cost2, ..., timeN costN, where timeN is the time that N-th clock is showing (so Rabbit can buy this time), and costN is the price of N-th clock's time. The second line stores Rabbit’s budget in cents (capacity). `output.txt` should contain the only number: maximum number of seconds that the White Rabbit can buy.

Example

`input.txt`

```
5 1 4 2 3 3 2 4 1 5
6
```

`output.txt`

```
12
```