More completion about the code test.

I did not finish all tasks since the tasks increase one every day, I have to work in daytime and did not have time to finish all.

For time reason, some data members in some classes are directly access and not provided setter and getter which are not allowed actually.

Some check should be done more strictly in some tasks in reality but not implemented for time reason as well.

Here I will list all tasks key point I have done currently.

**Day One task**:

This task key point is how to quickly find two elements summary equals some value.

Firstly I sort this array, then set two pivots from most left index and most right index. Moving one of them according if these two elements do not equal the result.

**Day Two task**:

The key point is to parse the input text file, get the password policy, count the letters and check if they are in the scope of this policy

**Day Three task**:

This task is simple. Just make sure to get the right character when moving right 3, down 1

**Day Four task**:

The key point of this task is how to skip the blank line and construct the correct passport for each person.

**Day Five task**:

We can borrow some kind of thinkings from binary search. Narrow the search interval in half: 128->64->32->16->8->4->2->1.

**Day Six task**:

The key point is to count unique yes characters in a group without duplication.

**Day Seven task**:

It is a graph issue I think. Every bag represents a vertice in the graph. The contain relationship represents edge in graph. Since it is a containing and contained relationship. It is directed graph here.

We can use adjacency list to represent this graph and then do a depth first search(DFS) from the “shiny gold”, all vertice appear in the DFS are contained by “shiny gold” bag, so we can get the bags numbers which can contain “shiny gold” bag just using the whole number of vertice subtract the number appear in DFS of “shiny gold”

**Day Eight task:**

The key point of this task is find out when an instruction is accessed a second time. A flag indicates if an instruction is accessed already in Instruction class.

Another key point is when executing an instruction, we must pay attention to the steps is + or -.

In my code, for simplicity, I did not implement read the file, find all kinds of bags. I think this is not key point of this task.

Improvement in the future.

1. Improve arithmetic for some tasks
2. Decouple some methods as much as possible in some class.
3. Improve the code robust(doing more check for some parameters)
4. Split classes in different files in same task.
5. ……………..