EE4, B04901158
Jing-Ping Cheng

Data Structure and Advanced Programming
Homework 5

第一題

1. BDE; 2. CD

第二題

(a) `getEntry(position)`

```
4   template <typename Itemtype>
5   Itemtype LinkedList<Itemtype>::getEntry(int position) noexcept(false)
6   {
7       Itemtype node;
8       LinkedList<Itemtype>* aList = new LinkedList;
9       int cnt = this->getLength();
10      if (position >= 1 && position <= cnt) {
11          // shallow copy until getting the entry at position
12          // since always insert at the beginning of aList
13          // it is a stable insertion
14          for (int i = cnt; i >= position; i--) {
15              aList->insert(1, this->getLastEntry());
16              this->remove(i);
17          }
18
19          // since stable insertion,
20          // the entry of position is at the beginning
21          // keep insert at position, it is also a stable insertion
22          for (int i = position; i < cnt - 1; i++) {
23              this->insert(position, aList->getLastEntry());
24              aList.remove(aList->getLastEntry());
25          }
26          node = aList->getLastEntry();
27
28          this->insert(position, aList->getLastEntry());
29          aList.remove(aList->getLastEntry());
30          delete aList;
31          return nodePtr;
32      } else
33          throw(logic_error("invalid position!"));
34  }
```

(b) `class stack`

the following four functions, and the constructor, destructor

```
1   isEmpty():
2       return this->list.getLength() == 0;
3
4   push(elmnt):
5       int pos = this->list.getLength() + 1;
6       this->list.insert(pos, elmnt);
7
8   pop():
9       this->list.remove(this->list.getLength());
10
11  peek():
12      return this->list.getLastEntry();
```

(c) `stack::top()`

as shown at line11-12 in (b).

(d) `stack::push(elmnt), stack::pop()`

as shown at line 4-9 in (b).