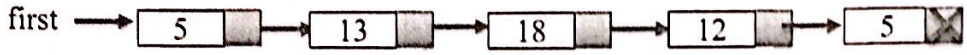
CSC 402 SP19 Homework 4 Part II name Ximan Liu

Write the answers to these problems on paper. Scan the paper and upload to the submissions folder.
We will grade a random subset of these for credit.

1. The Playground class (algs13) represents a linked-list of doubles, illustrated below.



An instance could be created as: Playground hw4 = new Playground();
Assume that the instance is somehow populated with data (as in the above figure).

Write an instance method of the Playground class that would compute the difference between the maximum and minimum values of a Playground instance. (the answer for the above figure would be 18-5 = 13)

a) Show how to call your function; use hw4 as the invoking instance.

```
hw4.difference( );
```

b) Write the function here.

```
Public double difference( ) {
    double max = first.item;
    double min = first.item;
for(Node temp = first; temp != null; temp = temp.next) {
    if (max < temp.item) {
        max = temp.item;
    }
    if (min > temp.item) {
        min = temp.item;
    } return (max-min);
}
```

Reference Example

The pseudocode below would reverse the contents of the stack s1. (assume the stack & queue store ints).

```
Stack s1 = new Stack();   // assume s1 is populated with data somehow
Queue q1 = new Queue();

while ( ! s1.isEmpty() ) {  // move stack contents to q1
     int item = s1.pop();
     q1.enqueue( item );
}
while ( ! q1.isEmpty()  ) {  // move queue contents to s1
   int item = q1.dequeue();
   s1.push( item )
}
```

Vers 1.0

- Over -

Ximan Liu

Assume stack s1 and queue q1 have been populated with data.
Write client pseudocode (using any combination of extra stacks & queues) to solve the following:

1. Reverse the queue q1

```
Stack s2 = new Stack( );
while (! q1. isEmpty( )) {   // move queue contents to s2;
    int item = q1. dequeue( );
    s2. push (item);
}
while(! s2. isEmpty( )) {   // move stack contents to q1;
    int item= s2. pop( );
    q1. enque (item);
}
```

2. Remove all even numbers from the stack s1

```
Stack s3 = new Stack( );
while (! s1. isEmpty( )) {
    int item = s1. pop( );
    if(item % 2 !=0 ) {
        s3. push(item);
    }
} while (! s3. isEmpty){
    s1. push (s3. pop( ));
}
```

3. *Nondestructively* make a copy of q1. Name the copy q2. (Nondestructively means that q1 is the same before and after the copy is made).

```
Queue q2 = new Queue( );
Queue q3 =  new Queue( );
while (! q1. isEmpty( )) {
    int item= q1. dequeue( );
    q2. enqueue(item);
    q3. enqueue(item);
}
while (! q3. isEmpty( )) {
    int item2 = q3. dequeue( );
    q1. enqueue (item2);
}
```

Vers 1.0

- Over -