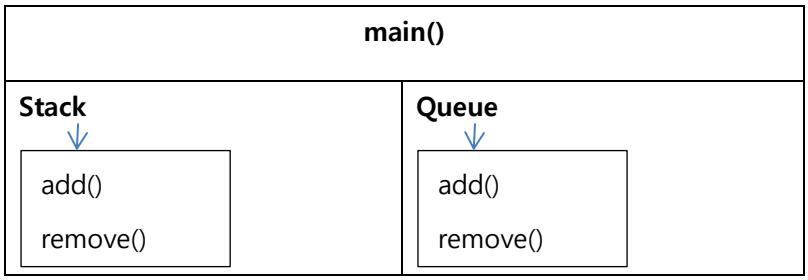
Mod2 (Simple Testing plan and Test)

Woohyuk Yang

yangwo@oregonstate.edu

※Test Plan and Testing Result



I illustrated the structure of the program to test the program how exactly it works with my intention.

1. Stack like structure testing plan and test

As stack structure is FILO(First in, Last out). So I planned to test adding few characters. and to check out it is added well and removed well. I planned to test printing characters in an order. So my intention was when I add ‘a’. ‘b’ ,’c’, and ‘d’ in an order. And stack structure is FILO structure so I guessed if I remove 4 characters and print it respectively, the program would print ‘d’, ‘c’, ‘b’ , and ‘a’, which is in reversed way.

```
/* stack structure test*/
   stack.add('a');
   stack.add('b');
   stack.add('c');
   stack.add('d');

   cout << stack.remove() << endl;
   cout << stack.remove() << endl;
   cout << stack.remove() << endl;
   cout << stack.remove() << endl;
```

```
flip3 ~/cs162/module/mod2_1 165% Mod2
d
c
b
a
```

Example code above shows that the program worked exactly as I intended.

I also tested when the stack is empty. When stack is empty, remove function can not remove a chracter

```
/* stack structure test*/
    stack.add('a');
    stack.add('b');
    stack.add('c');
    stack.add('d');

    cout << stack.remove() << endl;
    cout << stack.remove() << endl;
    cout << stack.remove() << endl;
    cout << stack.remove() << endl;
    cout << stack.remove() << endl;
    cout << stack.remove() << endl;
```

```
flip3 ~/cs162/module/mod2_1 165% Mod2
d
c
b
a
The stack is empty.

The stack is empty.
```

I removed 6 times as above after adding only 4 characters. So the result is like the one on the right above. The program prints out “The stack is empty.” Because I only added 4 letters but tried to remove 6 letters.

2. Queue like structure testing plan and test

As queue structure is FIFO(First in, First out). So I planned to test adding few characters. and to check out it is added well and removed well. I planned to test printing characters in an order. So my intention was when I add ‘a’. ‘b’ ,’c’, and ‘d’ in an order. And queue structure is FIFO structure so I guessed if I remove 4 characters and print it respectively, the program would print ‘a’, ‘b’, ‘c’ , and ‘d’, which is in an order I added to the queue before, which is different from FILO; stack.

```
/* queue structure test*/
    queue.add('a');
    queue.add('b');
    queue.add('c');
    queue.add('d');

  cout <<  queue.remove()<< endl;
  cout <<  queue.remove()<< endl;
  cout <<  queue.remove()<< endl;
  cout <<  queue.remove()<< endl;
```

```
a
b
c
d
```

Example code above shows that the program worked exactly as I intended.

I also tested when the queue is empty. When queue is empty, remove function can not remove a chracter

```
/* queue structure test*/
    queue.add('a');
    queue.add('b');
    queue.add('c');
    queue.add('d');

  cout <<  queue.remove()<< endl;
  cout <<  queue.remove()<< endl;
  cout <<  queue.remove()<< endl;
  cout <<  queue.remove()<< endl;
  cout <<  queue.remove()<< endl;
  cout <<  queue.remove()<< endl;
```

```
a
b
c
d
the queueu is empty

the queueu is empty
```

I removed 6 times as above after adding only 4 characters. So the result is like the one on the right above. The program prints out “The queue is empty.” Because I only added 4 letters but tried to remove 6 letters.