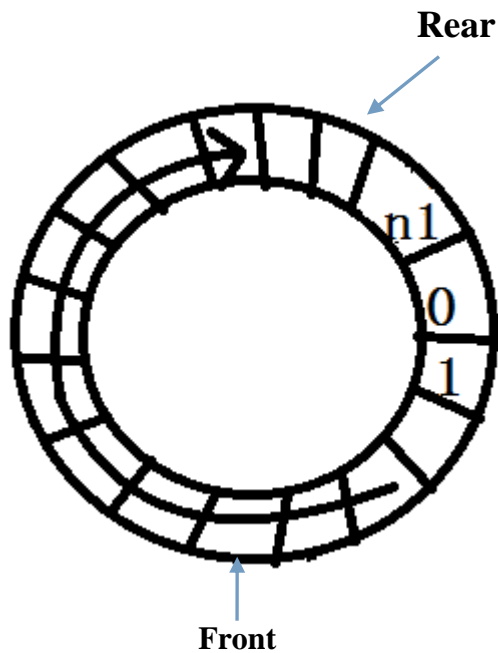


Assignment 6
CS 590
Stevens Institute of Technology
Week 7

1 Written Problems

1.1 Counting your Chickens

Solutions:



Above is a model of a queue in a circular array $[0 \dots n1]$.

So a formula for the number of elements can be:

NumberOfElements = rear - front + 1 ; if rear \geq front;

Or = rear - front + n1 ; if rear < front;

1.2 Pops and Pushes

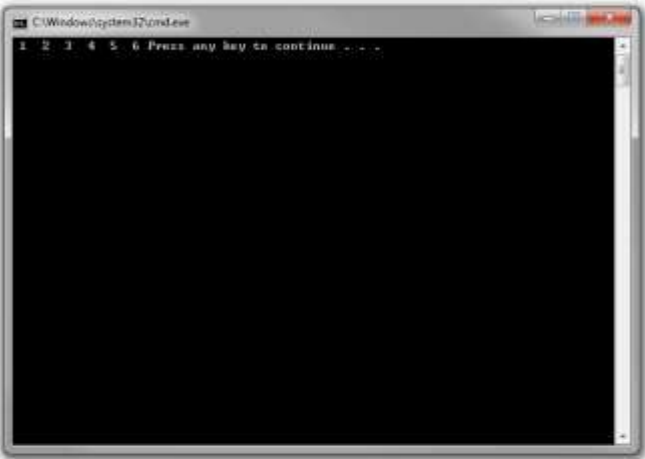
Solutions:

The sequences can be realized as the output of the stack which has the input stream:1,2,3,4,5,6 are (a), (b), (c).

For sequences (a):

Operation is :push pop push pop push pop push pop push pop push pop, the picture below is the result.

```
1 #include <iostream>
2 #include <stack>
3
4 int main()
5 {
6     std::stack<int> mystack;
7
8     mystack.push(1);
9     std::cout << " " << mystack.top() << " ";
10    mystack.pop();
11
12    mystack.push(2);
13    std::cout << " " << mystack.top() << " ";
14    mystack.pop();
15
16    mystack.push(3);
17    std::cout << " " << mystack.top() << " ";
18    mystack.pop();
19
20    mystack.push(4);
21    std::cout << " " << mystack.top() << " ";
22    mystack.pop();
23
24    mystack.push(5);
25    std::cout << " " << mystack.top() << " ";
26    mystack.pop();
27
28    mystack.push(6);
29    std::cout << " " << mystack.top() << " ";
30    mystack.pop();
31
32    return 0;
33 }
```



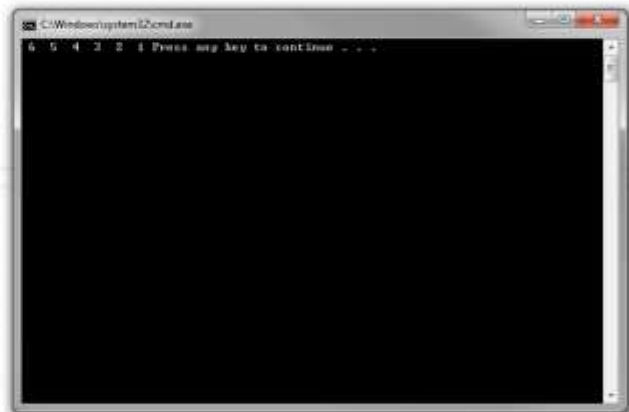
For sequences (b):

Operation is :push push push push push push pop pop pop pop pop pop, the picture below is the result.

```

1 #include <iostream>
2 #include <stack>
3
4 int main()
5 {
6     std::stack<int> mystack;
7
8     mystack.push(1);
9     mystack.push(2);
10    mystack.push(3);
11    mystack.push(4);
12    mystack.push(5);
13    mystack.push(6);
14
15    std::cout << " " << mystack.top() << " ";
16    mystack.pop();
17    std::cout << " " << mystack.top() << " ";
18    mystack.pop();
19    std::cout << " " << mystack.top() << " ";
20    mystack.pop();
21    std::cout << " " << mystack.top() << " ";
22    mystack.pop();
23    std::cout << " " << mystack.top() << " ";
24    mystack.pop();
25    std::cout << " " << mystack.top() << " ";
26    mystack.pop();
27
28    return 0;
29 }

```



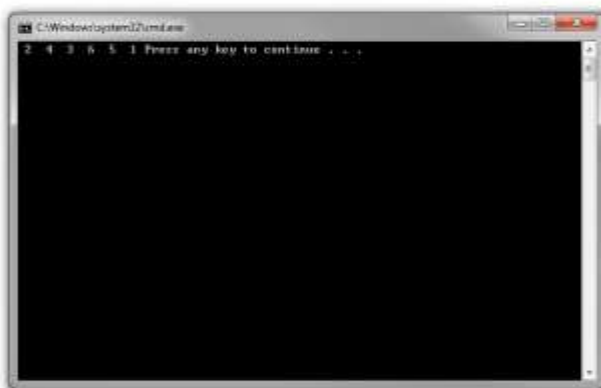
For sequences (c):

Operation is :push push pop push push pop pop push push pop pop pop, the picture below is the result.

```

1 #include <iostream>
2 #include <stack>
3
4 int main()
5 {
6     std::stack<int> mystack;
7
8     mystack.push(1);
9     mystack.push(2);
10    std::cout << " " << mystack.top() << " ";
11    mystack.pop();
12
13    mystack.push(3);
14    mystack.push(4);
15    std::cout << " " << mystack.top() << " ";
16    mystack.pop();
17    std::cout << " " << mystack.top() << " ";
18    mystack.pop();
19
20    mystack.push(5);
21    mystack.push(6);
22    std::cout << " " << mystack.top() << " ";
23    mystack.pop();
24    std::cout << " " << mystack.top() << " ";
25    mystack.pop();
26    std::cout << " " << mystack.top() << " ";
27    mystack.pop();
28
29    return 0;
30 }

```



2 Programming Problems

The solutions are attached as cpp and java files.