Part 1

Assume a normal queue that can accommodate maximum six elements. At a given instance, front = 2 and rear = 4 and the queue elements look like this:

`(),(),L,M,N,()`

What will be the value of front and rear if 'O' is inserted into the queue? Explain your answer.

Given that currently the list points to the last full element it must be assumed that is how this list is implemented. Therefore adding the 'O' would make front = 2 and rear = 5.

Part 2

What does the following code snippet output? Explain.

```
#include <iostream>
#include <queue>
using namespace std;

int main() {
  queue<int> q;
  for (int i = 1; i <= 5; ++i) {
    q.push(i);
  }
  cout << "Initial size: " << q.size() << endl;
  for (int i = 0; i < 3; i++) {

    q.push(q.front());
    q.pop();
  }

  cout << q.front() << " ";

  cout << "\nSize at end: " << q.size() << endl;
  return 0;
}
```

stdout:

```
Initial size: 5
4
Size at end: 5
```

This code makes a queue with the values 1 through 5 in it. It then prints the size which is 5. Then it moves the first element to the back. This process involves removing one element and adding another so the size is the same. 3 elements are moved to the back (1, 2, 3) so the first element is 4 which is why it is what is printed. As mentioned before the size doesn't change so it still outputs 5.