

23 September 2021

1. **Warm up:** Answer the following True / False questions.
 - (a) A doubly linked list of 10 integers take up twice the space of a singly linked list of 10 integers.
 - (b) A circularly linked list can do the same things as a doubly linked list.
 - (c) Positive integers support iterators.
2. This question is about **queues**, implemented using an array in a wrap-around method. The queue has size 10, and the **enqueue** and **dequeue** functions are executed randomly.
 - (a) If these functions are executed 6 times, what are the possible lengths of the queue afterward? What is the average length?
 - (b) Suppose the array holding the queue has size 14, and **enqueue** has a probability of $\frac{3}{4}$ being executed, and **dequeue** has a probability of $\frac{1}{4}$ being executed.
 - i. What is the probability the queue will be full after 6 executions? The queue is full if one spot in the array is empty.
 - ii. **Bonus:** What is the expected value for number of executions at which the queue is full?

The next two problems refer to the following uncompiled C++ files.

lists.cpp

```

1  #include <list>
2  #include <iostream>
3  using namespace std;
4  int main() {
5      list<int> L({ 1, 2, 3, 4, 2});
6      int i = 0;
7      for (list<int>::iterator a = L.begin(); a != L.end(); a++) {
8          int avalue = *a;
9          if (avalue < 3) {
10             L.push_back(i);
11         }
12         i++;
13     }
14     for (int l: L) {
15         cout << l << " ";
16     }
17     cout << endl;
18     return 1;
19 }
```

chop.cpp

```

1  #include <list>
2  #include <iostream>
3  #include <string>
4  using namespace std;
5
6  list<T> chop (list<T> input, int length) {
7      ...
8  };
9
10 int main() {
11     list<int> L1({ 2, 2, 5, 5, 1});
12     chop(L1, 4); // list {2, 2, 5, 5} is returned
13     chop(L1, 10); // list {2, 2, 5, 5, 1} is returned
14     chop(L1, -1); // list {2, 2, 5, 5, 1} is returned
15     list<string> L2({ "hey", "whey", "they", "day", "cray"});
16     chop(L2, 2); // list {"hey", "whey"} is returned
17     chop(L2, 0); // list {} is returned
18 }
```

3. This question is about the program that results from compiling `lists.cpp`.
 - (a) What will be the output?
 - (b) Suppose line 10 is changed to `L.push_front(i);`. What will be the output?
 - (c) Suppose line 10 is changed to `L.push_back(avalue);`. What will be the output?
 - (d) If there are n entries in list `L` on line 5, what is the largest possible size of `L` after line 13?
4. This question is about the program that results from compiling `chop.cpp`.
 - (a) Fill in the code on line 7 to make the program work as expected.
 - (b) Suppose the argument `int start` is added to `chop`, indicating the index of the list at which the returned list should start (it should still have the length `length`). Adjust the body of `chop` accordingly.