Name: ________________________________

---

1. (5 points) Determine the stack contents at the points indicated below during the following operations on the stack ADT. Write down the stack contents after the operation on the given line is executed. Be sure to indicate the top of the stack.

```
1     stack<int> s;
2     s.push(9)
3     s.push(5)
4     s.push(10)
5     s.push(1)
6     s.pop()
7     s.pop()
8     s.push(0)
9     s.pop()
10    s.pop()
11    s.push(3)
```

After line 1:

After line 3:

After line 5:

After line 8:

After line 11:

Name: ________________________________

2. (5 points) Determine the queue contents at the points indicated below during the following operations on the queue ADT. Write down the queue contents after the operation on the given line is executed. Be sure to indicate the front and back of the queue.

```
1     queue<int> q;
2     q.enqueue(5);
3     q.dequeue();
4     q.enqueue(12);
5     q.enqueue(45);
6     q.enqueue(1);
7     q.dequeue();
8     q.dequeue();
9     q.enqueue(100);
10    q.enqueue(42);
11    q.dequeue();
```

After line 1:

After line 3:

After line 5:

After line 8:

After line 11:

Name: ________________________________

3. (5 points) Determine the deque contents at the points indicated below during the following operations on the queue ADT. Write down the deque contents after the operation on the given line is executed. Be sure to indicate the front and back of the deque.

```
1     deque<int> dq;
2     dq.enqueue_front(9);
3     dq.dequeue_front();
4     dq.enqueue_front(1);
5     dq.enqueue_back(42);
6     dq.enqueue_front(1);
7     dq.enqueue_front(17);
8     dq.dequeue_back();
9     dq.enqueue_back(39);
10    dq.enqueue_back(6);
11    dq.dequeue_front();
```

After line 1:

After line 3:

After line 5:

After line 8:

After line 11:

Name: ________________________________

4. (8 points) Consider a priority queue implemented as a max heap using an array of size 6 with methods enqueue(entry, priority) and dequeue(). Determine the priority queue contents, represented both an array and as a tree, at the points indicated below during the following operations on the ADT. Write down the queue contents after the operation on the given line is executed.

```
1     priority_queue<string, int> pq;
2     pq.enqueue("a", 5);
3     pq.dequeue();
4     pq.enqueue("b", 12);
5     pq.enqueue("c", 45);
6     pq.enqueue("d", 1);
7     pq.dequeue();
8     pq.dequeue();
9     pq.enqueue("e", 100);
10    pq.enqueue("f", 42);
11    pq.enqueue("g", 504);
```

After line 6:

array view   tree view

After line 11:

array view   tree view

Name: ________________________________

5. (5 points) Determine the binary tree(s) contents at the points indicated below during the following operations on the binary tree ADT. Write down the tree structure after the operation on the given line is executed. Be sure to indicate the root of each tree.

```
1     BinaryTree<int> t1(4);
2     BinaryTree<int> t2(12);
3     BinaryTree<int> t3(1);
4     BinaryTree<int> t4(62);
5     t2.attachRightSubtree(t3);
6     t1.attachRightSubtree(t2);
7     t4.attachLeftSubtree(t1);
8     t4.detachLeftSubtree();
```

After line 4:

After line 5:

After line 6:

After line 7:

After line 8:

Name: ________________________________

6. (4 points) Consider a binary search tree (with no balancing) containing integer keys. Write down the tree structure after the operation on the given line is executed. Be sure to indicate the root of the tree.

```
1     BST<int> t;
2     t.insert(51);
3     t.insert(102);
4     t.insert(12);
5     t.insert(75);
6     t.remove(51);
7     t.insert(18);
8     t.remove(12);
```

After line 5:

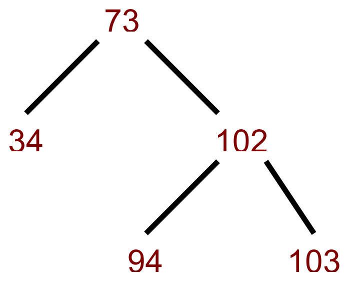
After line 6:

After line 7:

After line 8:

Name: ________________________________

7. (4 points) Consider the following Binary Search Tree with integer keys



a) sketch the tree structure after performing a left rotation about the node with key = 73

b) sketch the tree structure after performing a right rotation about the node with key = 102 on the tree resulting from part a)

Name: ________________________________

8. (8 points) Consider a binary search tree, balanced as a treap, containing integer keys. Write down the tree structure after the operation on the given line is executed. Be sure to indicate the root of the tree. A randomly generated priority for each insert is given.

```
1     BST<int> t;
2     t.insert(51);  // use a priority of 8
3     t.insert(102); // use a priority of 92
4     t.insert(12);  // use a priority of 3
5     t.insert(300);  // use a priority of 27
6     t.remove(102);
7     t.insert(18);  // use a priority of 12
8     t.remove(12);
```

After line 5:

After line 6:

After line 7:

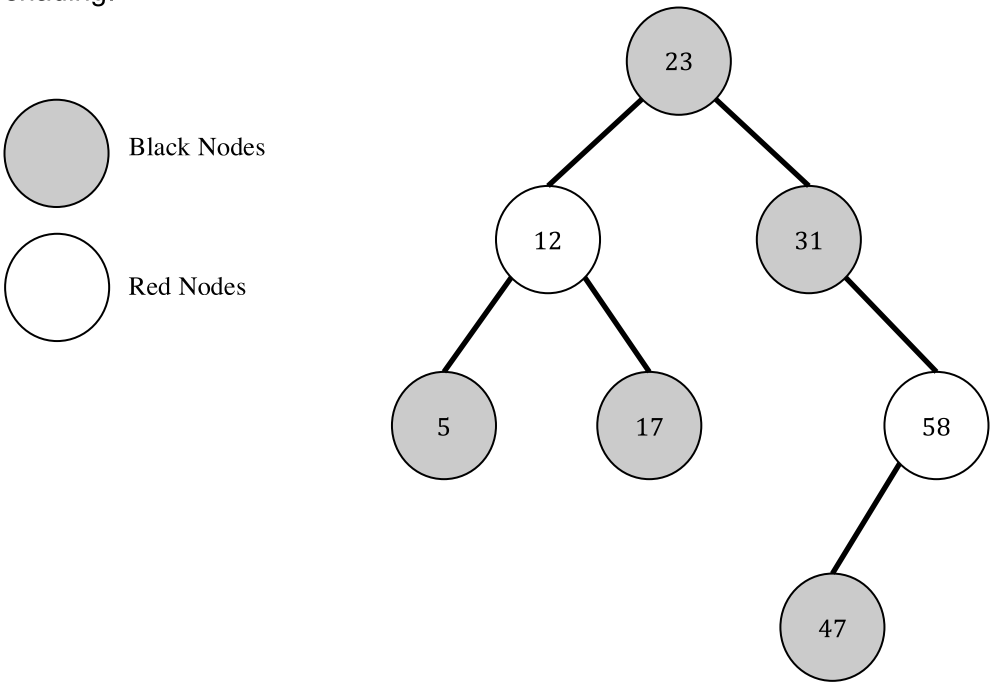
After line 8:

Name: ______________________________

9. (5 points) Consider a hash table of length 13 using *linear probing* and a hash function h(k) = k mod 13. Determine the hash table contents after the following insertions: 4, 8, 14, 21, 30, 17. Sketch the table.

Name: ________________________________

10. (6 points) Consider the following red-black tree with black nodes indicated by shading.



a) Add the augmented nodes on the tree above.

b) Is this a valid red-black tree? Be sure to indicate your reasoning.

Valid red-black tree: TRUE or FALSE

Reasoning:

Name: ________________________________

11. (12 points)

a) Sketch the following directed graph G with vertices {A, B, C, D, E} and edges represented as the following adjacency matrix.

	A	B	C	D	E
A	0	1	1	0	0
B	0	0	0	1	1
C	0	0	0	0	0
D	0	0	0	0	0
E	0	0	0	0	0

b) What order are nodes visited (printing left-right) in a depth-first search rooted at node A? Nodes are expanded in alphabetical order.

c) What order are nodes visited (printing left-right) in a breadth-first search rooted at node A? Nodes are expanded in alphabetical order.

Name: ________________________________

12. (7 points) You are presented with a problem that requires storing a small (< 100) number of objects one time, and then searching for objects many times. No items are inserted or deleted during searches. You do not need to quickly visit each object in sorted key order and you know all possible objects you will encounter ahead of time.

What data structure would you use?

Briefly justify your selection:

13. (6 points) Short answer

a) What are some advantages of templates in C++?

b) How does one define an interface for an ADT in C++?

c) What are the advantages of arrays over linked-lists when implementing linear data structures?

Name: ________________________________

14. (10 points) Consider the following partial C++ code representing a **circular** doubly-linked list, in which the last node links back to the first and vice-versa.

```
class List
{
    struct Node
    {
      int data;
      Node * next;
      Node * prev;
    };

    Node head;
};
```

(Note the variable `head` is not dynamically allocated)

a) Write a default constructor for the class List.

b) Write a method for the class List to test if the List is empty.

Name: ________________________________

15. (10 points) Consider the mergesort algorithm applied to the following array of integers: {93,64,21,82,12,13}. Sketch the sub-arrays input at each level of the recursion as well as the returned sub-arrays at each level.