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<string> s;
2     s.push("a")
3     s.push("z")
4     s.push("zorg")
5     s.push("zero")
6     s.pop()
7     s.push("gift")
8     s.pop()
9     s.pop()
10    s.pop()
11    s.pop()
```

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(104);
3     q.enqueue(42);
4     q.enqueue(1);
5     q.enqueue(1000);
6     q.enqueue(-9);
7     q.dequeue();
8     q.dequeue();
9     q.enqueue(-84);
10    q.dequeue();
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_back(1);
3     dq.enqueue_front(42);
4     dq.enqueue_front(24);
5     dq.enqueue_back(7);
6     dq.enqueue_front(1);
7     dq.enqueue_back(59);
8     dq.dequeue_back();
9     dq.dequeue_front();
10    dq.dequeue_back();
11    dq.dequeue_back();
```

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 a 1-based array of size 8 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.enqueue("b", 1);
4     pq.enqueue("b", 42);
5     pq.enqueue("d", 45);
6     pq.dequeue();
7     pq.dequeue();
8     pq.enqueue("r", 7);
9     pq.enqueue("q", -1);
10    pq.enqueue("t", 46);
11    pq.enqueue("g", -9);
```

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(s) after the operation on the given line is executed. Be sure to indicate the root of each tree.

```
1     BinaryTree<int> t1(78);
2     BinaryTree<int> t2(46);
3     t1.attachLeftSubtree(t2);
4     BinaryTree<int> t3(57);
5     BinaryTree<int> t4(105);
6     t1.attachRightSubtree(t4);
7     BinaryTree<int> t5(57);
8     t3.attachLeftSubtree(t5);
9     t1.detachRightSubtree();
```

After line 5:

After line 6:

After line 7:

After line 8:

After line 9:

Name: _____

6. (7 points) Consider a binary search tree (with no balancing) containing integer keys and string values. 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,string> t;
2     t.insert(51, "foo");
3     t.insert(12, "bar");
4     t.remove(51);
5     t.insert(51, "foo");
6     t.insert(109, "bar");
7     t.insert(1, "gift");
8     t.remove(12);
```

After line 5:

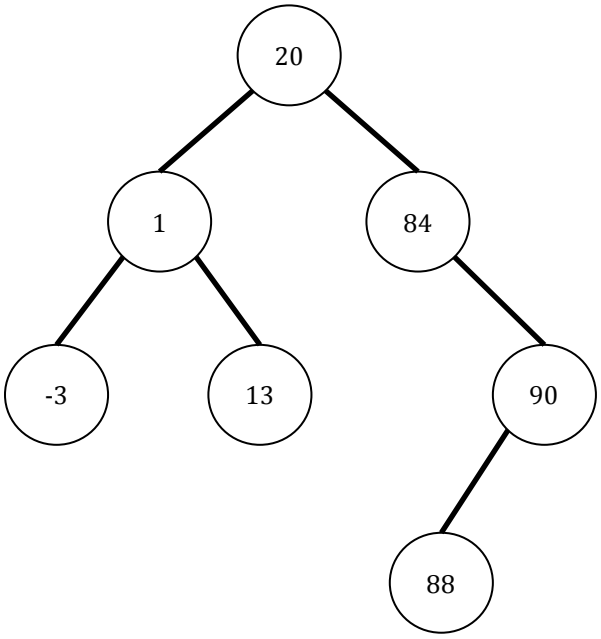
After line 6:

After line 7:

After line 8:

Name: _____

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



a) sketch the tree structure after performing a left rotation about the node with key = 84

b) sketch the tree structure after performing a right rotation about the node with key = 20 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(104);  // use a priority of 8
3     t.insert(62); // use a priority of 64
4     t.insert(12);  // use a priority of 109
5     t.insert(1);  // use a priority of 32
6     t.remove(104);
7     t.insert(42);  // use a priority of 7
8     t.remove(12);
```

After line 5:

After line 6:

After line 7:

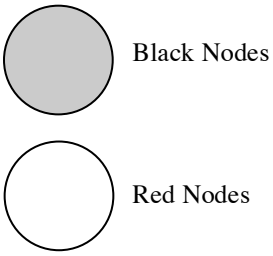
After line 8:

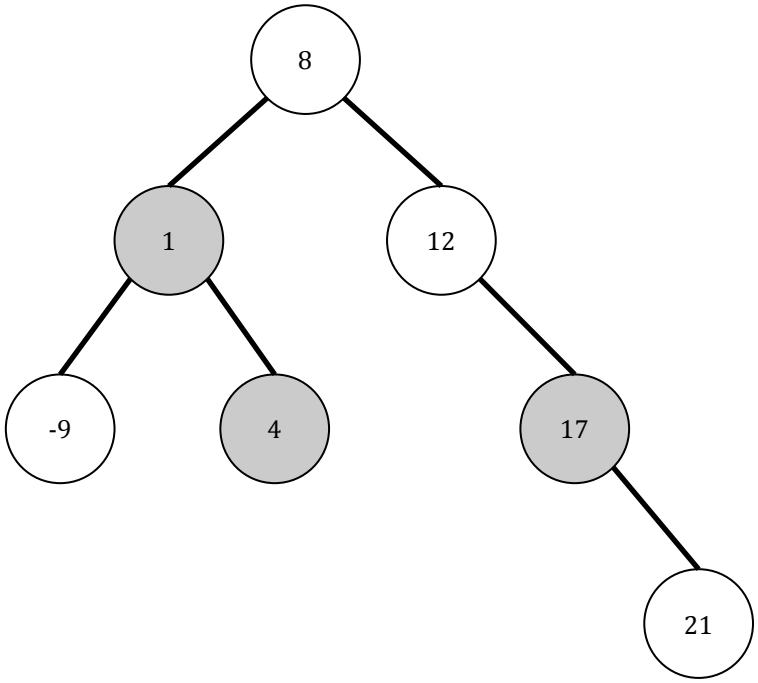
Name: _____

9. (5 points) Consider a hash table containing integer keys with a length of 13 using *quadratic probing* and a hash function h(k) = k mod 13. Determine the hash table contents after the following insertions: 13, 94, 62, 16, 29, 12. 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	0	1	1
B	0	0	1	0	1
C	0	0	0	1	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 possibly large (> 10 million) number of objects one time, and then searching for objects many times, removing them each time they are found. You need to be able to write the objects, in sorted order, to a file as fast as possible.

What data structure would you use?

Briefly justify your selection:

13. (10 points) Consider the following partial C++ code representing a singly-linked list holding integer values.

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

    Node *head, *tail;
};
```

a) Write a default constructor for the class List.

Name: _____

13. cont.

b) Write a method with signature `void append(int item)` to append item to the end of the list. Assume all allocations succeed.

Name: _____

14. (10 points) Consider an algorithm that for an input of size N, inserts each value into Red-Black Tree. What is the overall best-case and worst-case complexity of this algorithm with respect to the number of comparisons? Justify your reasoning.