

Q1

- 1 `mystery(4, 9)` returns $4 + \text{mystery}(8, 4) = 4 + 32 + 0 = 36$
- 2 `mystery(8, 4)` returns `mystery(16, 2)`
- 3 `mystery(16, 2)` returns `mystery(32, 1)`
- 4 `mystery(32, 1)` returns $32 + \text{mystery}(64, 0)$
- 5 `mystery(64, 0)` returns 0

This code returns the product of a and b (it multiplies them)

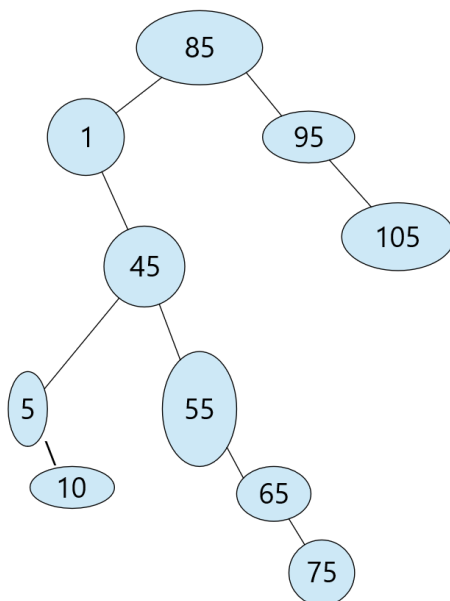
Q2

0	1	2	3	4	5	6	7	8	9	10		Left	right	X	(mid)		returns(array, 0, 4, 4)
7	9	12	43	50	51	60	61	62	64	76		0	4		4	5	
0	1	2	3	4	5	6	7	8	9	10		Left	right	X	(mid)		returns(array, 0, 2, 4)
7	9	12	43	50	51	60	61	62	64	76		0	2		4	2	
0	1	2	3	4	5	6	7	8	9	10		Left	right	X	(mid)		returns(array, 0, -1, 4)
7	9	12	43	50	51	60	61	62	64	76		0	-1		4	0	
																	returns -1

Q3

Binary search is better because it is much faster. Instead of having to check each element of the array, a very large number of checks can be skipped.

Q4



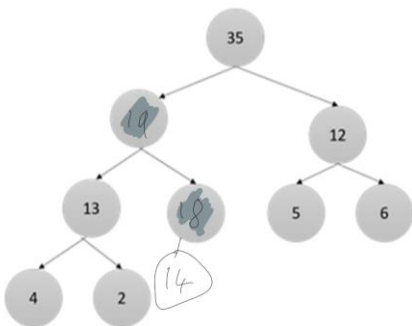
Q5

```
int getDepth(Node *root)
{
    if (root == NULL) return 0;
    int leftResult, rightResult;
    leftResult = getDepth(root->left);
    rightResult = getDepth(root->right);
    if (leftResult >= rightResult)
    {
        return 1+leftResult;
    } else
    {
        return 1+rightResult;
    }
}
```

Q6

The tree is not a Max Heap, because heaps are populated from left to right.

Q7



Q8

