Q1

- 1 mystery(4, 9) returns 4 + 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 + mystery(64, 0)
- 5 mystery(64, 0) returns 0

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

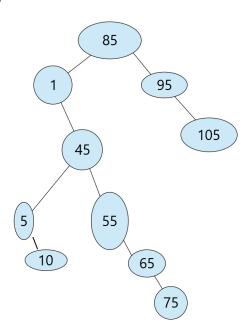
Q2

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

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

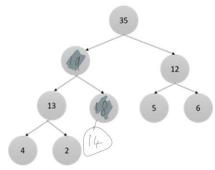


Q5

```
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

