```
int main(){

// Starting at row 2, column 1 from left to right on "Catalog of tile types".
// Lake Features

        Feature * f1  = new Lake(false, 0);
        Feature * f2  = new Lake(false, 1);
        Feature * f3  = new Lake(false, 2);
        Feature * f4  = new Lake(false, 2);
        Feature * f5a = new Lake(true, 1);
        Feature * f5b = new Lake(true, 1);
        Feature * f6  = new Lake(true, 3);
        Feature * f7a = new Lake(true, 1);
        Feature * f7b = new Lake(true, 1);
        Feature * f8  = new Lake(true, 3);
        Feature * f9  = new Lake(true, 3);
        Feature * f10 = new Lake(true, 3);
        Feature * f11 = new Lake(true, 3);
        Feature * f12 = new Lake(true, 3);
        Feature * f13 = new Lake(true, 3);
        Feature * f14 = new Lake(false, 1);
        Feature * f15 = new Lake(true, 3);
        Feature * f16 = new Lake(true, 3);
        Feature * f17 = new Lake(false, 2);
        Feature * f18 = new Lake(false, 2);
        Feature * f19 = new Lake(true, 3);
        Feature * f20 = new Lake(true, 3);
        Feature * f21 = new Lake(false, 1);

// Merging multiple combinations of 'lake' tiles to check whether or not a tiger can be placed.
//Test: Placing Crocodile and Checking Score
```

```
// Test Case 1
        /*f1->merge(f14);
        f1->addCrocodile(1);
        int a=0;
        int b=0;
        f1->canPlaceCrocodile();
        f2->canPlaceCrocodile();
        f1->checkIfScored(a,b);
        a=0;
        b=0;
        f2->checkIfScored(a,b);

// Test Case 2
        f6->merge(f7b);
        f6->addCrocodile(1);
        f6->canPlaceCrocodile();
        f7b->canPlaceCrocodile();
        f6->checkIfScored(a,b);
        f7b->checkIfScored(a,b);

// Test Case 2
        f10->merge(f12);
        f10->addTiger(1);
        f10->canPlaceCrocodile();
        f12->canPlaceCrocodile();
        f10->checkIfScored(a,b);
        f10->checkIfScored(a,b);*/

// Test Case 4
        /* f5b->merger(f15);
        f5b->addCrocodile(1);
        int a=0;
```

```
        int b=0;
        f5b->canPlaceCrocodile();
        f15->canPlaceCrocodile();
        f5b->checkIfScored(a,b);
        a=0;
        b=0;
        f15->checkIfScored(a,b);

// Test Case 5
        f5a->merge(f12);
        f5a->addCrocodile(1);
        f5a->canPlaceCrocodile();
        f12->canPlaceCrocodile();
        f5a->checkIfScored(a,b);
        f12->checkIfScored(a,b);*/

// Test Case 6
        /*f4->merge(f8);
        f4->addCrocodile(1);
        int a = 0;
        int b = 0;
        f4->canPlaceCrocodile();
        f8->canPlaceCrocodile();
        f4->checkIfScored(a,b);
        a=0;
        b=0;
        f8->checkIfScored(a,b);

// Test Case 7
        f3->merge(f10);
        f3->addCrocodile(1);
        f3->canPlaceCrocodile();
```

```
        f10->canPlaceCrocodile();
        f3->checkIfScored(a,b);
        f10->checkIfScored(a,b);

// Test Case 8
        f9->merge(f13);
        f9->addCrocodile(1);
        f9->canPlaceCrocodile();
        f13->canPlaceCrocodile();
        f9->checkIfScored(a,b);
        f13->checkIfScored(a,b);*/

// Test Case 9
        /*f10->merge(f11);
        f10->addTiger(1);
        int a = 0;
        int b = 0;
        f10->canPlaceTiger();
        f11->canPlaceTiger();
        f10->checkIfScored(a,b);
        a=0;
        b=0;
        f11->checkIfScored(a,b);*/

// Test Case 10
        /*f21->merge(f18);
        f21->addCrocodile(1);
        int a = 0;
        int b = 0;
        f21->canPlaceCrocodile();
        f18->canPlaceCrocodile();
        f21->checkIfScored(a,b);
```

```
        a=0;
        b=0;
        f18->checkIfScored(a,b);

// Test Case 11
        f18->merge(f19);
        f18->addCrocodile(1);
        int a = 0;
        int b = 0;
        f18->canPlaceCrocodile();
        f19->canPlaceCrocodile();
        f18->checkIfScored(a,b);
        f19->checkIfScored(a,b);*/

// Test Case 12
        /*f20->merge(f2);
        f20->addCrocodile(1);
        int a =0;
        int b = 0;
        f20->canPlaceCrocodile();
        f2->canPlaceCrocodile();
        f20->checkIfScored(a,b);
        a=0;
        b=0;
        f2->checkIfScored(a,b);

// Test Case 13
        f16->merge(f2);
        f16->addCrocodile(1);
        f16->canPlaceCrocodile();
        f2->canPlaceCrocodile();
        f16->checkIfScored(a,b);
```

```
        f2->checkIfScored(a,b);


// Test Case 14
        f17->merge(f2);
        f17->addCrocodile(1);
        f17->canPlaceCrocodile();
        f2->canPlaceCrocodile();
        f17->checkIfScored(a,b);
        f2->checkIfScored(a,b);*/


};
```