2. Broken Economy.

arr → { 1 5 10 15 22 33 40 42 55 66 }

x = 34 ceil = 40 floor = 33

x = 42 ceil = 42 floor = 2

x = 12

{ c = 15

{ f = 10

ceil

floor.

arr → { 1 5 10 15 22 33 40 42 55 67 }

Approach

x = 34.

f = 22 33 40

c = 42

1. mid = 0+9/2 = 4

arr(mid) == x

22 > 34

34

2. mid = 5+9/2 = 7

42 > 34

42

34

3. mid = l+r/2 = 5

33 < 34

34

33

4. l = 6 } mid = 6.

r = 6

34 < 40

l ≤ r

x =

```
public static int firstIndex(int [] arr, int x){
    int li = 0;
    int ri = arr.length-1;
    int fi = -1;

    while(li<=ri){
        int mid = (li+ri)/2;

        if(arr[mid] == x){
            fi = mid;
            //left me dhundho bcoz firstIndex is always on left
            ri = mid - 1;

        }
        else if(arr[mid] > x){
            ri = mid - 1;
        }
        else{
            li = mid + 1;
        }

    }

    return fi;

}
```

x = 3

fi = 3

1. 0 ≤ 9 ✓

mid = 0+9/2 = 4

2. 0 ≤ 3 ✓

mid = 0+3/2 = 1

3. 2 ≤ 3 ✓

mid = 2+3/2 = 2

4. 3 ≤ 3 ✓

li ≤ ri ✓

mid = 3+3/2 = 3

arr[3] == x

3 == 3

li ≤ ri

```
public  static void brokenEconomy(int [] arr, int x){
    int li = 0;
    int ri = arr.length-1;

    int ceil = 0;
    int floor = 0;

    while(li<=ri){
        int mid = (li+ri)/2;

        if(arr[mid] == x){

            ceil = arr[mid];
            floor = arr[mid];
            break;
        }
        else if(arr[mid] > x){

            ceil = arr[mid];
            ri = mid-1;
        }
        else {

            floor = arr[mid];
            li = mid + 1;
        }

    }
    System.out.println(ceil);
    System.out.println(floor);
}
```

c = 55

f = 42

x = 43

mid = 4

mid = 7

43 > 42

mid = 8+9/2 = 8

found it

55 > 43

55

l ≤ r

1. First & last index -

arr →

x = 3

mid = 0+14/2 = 7

mid = 0+6/2 = 3

3 → 4

l ≤ r

mid = 6+4/2 = 5

3 → 5

if( )

{ x == arr(mid)

index → fi

}

first index

x = 3

1. m = 0+12/2 = 6

if arr(mid) == x

fi = 4

2. m = 0+5/2 = 2

3. mid = 3+5/2 = 4

mid = 0+3/2 = 1

mid = 2+3/2 = 2

mid = 3+3/2 = 3