

Question - 1

Buying Show Tickets

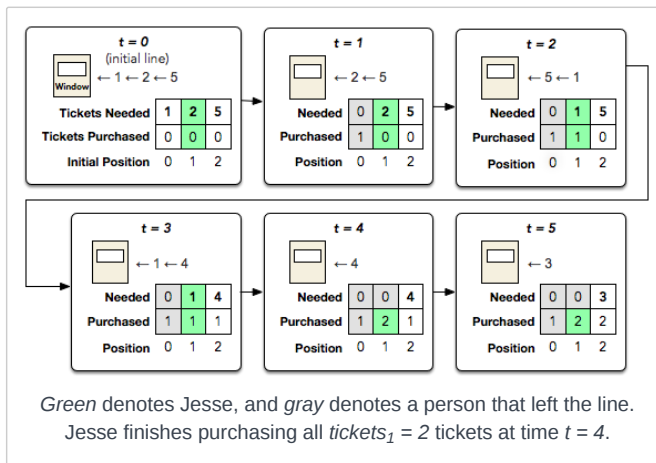
SCORE: 50 points

Easy Algorithms Arrays Core Skills Problem Solving

A line has formed to buy tickets for a concert. In order to delay a shortage caused by brokers buying large blocks of tickets, venue management has decided to sell only one ticket at a time. Buyers have to wait through the line again if they want to buy more tickets. Jesse is standing in line and has a number of tickets to purchase.

Given a list of ticket buyers with their numbers of desired tickets, determine how long it will take Jesse to purchase his tickets. Jesse's position in line will be stated, and each transaction takes 1 unit of time. For your purposes, no time is spent moving to the back of the line.

For example, if the zero indexed array of ticket requirements, $tickets = [1, 2, 5]$, and Jesse's position $p = 1$, the first five seconds of ticket sales look like this:



Function Description

Complete the function `waitingTime` in the editor below. The function must return an integer representing the units of time it takes Jesse to purchase his desired number of tickets.

`waitingTime` has the following parameter(s):

$tickets[tickets[0], \dots, tickets[n-1]]$: an array of tickets desired by each person at position $tickets[i]$

p : Jesse's position in line

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq tickets[i] \leq 10^9$, where $0 \leq i < n$.
- $0 \leq p < n$

► Input Format for Custom Testing

▼ Sample Case 0

Sample Input 0

```
5
2
6
3
4
5
2
```

Sample Output 0

```
12
```

Explanation 0

Given *tickets* = [2, 6, **3**, 4, 5], Jesse's position in line is marked in bold. His wait time looks like this:

0. *window* ← 2 ← 6 ← **3** ← 4 ← 5
1. *window* ← 6 ← **3** ← 4 ← 5 ← 1
2. *window* ← **3** ← 4 ← 5 ← 1 ← 5
3. *window* ← 4 ← 5 ← 1 ← 5 ← **2**
4. *window* ← 5 ← 1 ← 5 ← **2** ← 3
5. *window* ← 1 ← 5 ← **2** ← 3 ← 4
6. *window* ← 5 ← **2** ← 3 ← 4 (the person at the head of the line in the previous step purchased their last ticket and does not re-enter the line)
7. *window* ← **2** ← 3 ← 4 ← 4
8. *window* ← 3 ← 4 ← 4 ← **1**
9. *window* ← 4 ← 4 ← **1** ← 2
10. *window* ← 4 ← **1** ← 2 ← 3
11. *window* ← **1** ← 2 ← 3 ← 3
12. *window* ← 2 ← 3 ← 3 (Jesse purchased his last ticket and does not re-enter the line)

It took a total of 12 units of time to purchase 2 tickets.

► Sample Case 1

► Sample Case 2

Question - 2

SCORE: 5 points

Java

Easy

Algorithms

Sorting

What is the expected result?

```
import java.util.*;

public class VLA2 implements Comparator<VLA2> {
    int dishSize;

    public static void main(String[] args) {
        VLA2[] va = {new VLA2(40), new VLA2(200),
        new VLA2(60)};
        Arrays.sort(va, va[0]);
        int index = Arrays.binarySearch(va, new
        VLA2(40), va[0]);
        System.out.print(index + " ");
        index = Arrays.binarySearch(va, new
```

```
VLA2(80), va[0]);
    System.out.print(index);
}

public int compare(VLA2 a, VLA2 b) {
    return b.dishSize - a.dishSize;
}

VLA2(int d) { dishSize = d; }
}
```

- ☐ 0-2
- ☐ 0-3
- ☐ 2-1
- ☒ 2-2
- ☐ Compilation fails.
- ☐ An exception is thrown at runtime.

Question - 3

SCORE: 5 points

Java

Easy

Static Methods

What is the result of compiling and running the following program?

```
class test {
    public static void main(String args[]) {
        int[] arr = {1,2,3,4};
        call_array(arr[0], arr);
        System.out.println(arr[0] + "," +
arr[1]);
    }
    static void call_array(int i, int arr[]) {
        arr[i] = 6;
        i = 5;
    }
}
```

- ☐ 1,2
- ☐ 5,2
- ☒ 1,6
- ☐ 5,6

Question - 4

SCORE: 5 points

Java

Easy

Inheritance

Object Oriented Programming

What is the output for the below code ?

```
1. public class A {
2.     int add(int i, int j){
3.         return i+j;
```

```
4.     }
5. }
6. public class B extends A{
7.     public static void main(String argv[]){
8.         short s = 9;
9.         System.out.println(add(s,6));
10.    }
11.}
```

☐ Compile fail due to error on line no 2



Compile fail due to error on line no 9 due to the non-static method referenced from a static context.

☐ Compile fail due to error on line no 9 due to type mismatch.

☐ 15

Question - 5

SCORE: 5 points

Java

Medium

Interfaces

Object Oriented Programming

What is the expected result?

```
import java.util.*;

public class MyPancake implements Pancake {

    public static void main(String[] args) {
        List<String> x = new ArrayList<String>();
        x.add("3"); x.add("7"); x.add("5");
        List<String> y = new
MyPancake().doStuff(x);
        y.add("1");
        System.out.println(x);
    }

    List<String> doStuff(List<String> z) {
        z.add("9");
        return z;
    }
}

interface Pancake {
    List<String> doStuff(List<String> s);
}
```

☐ [3, 7, 5]

☐ [3, 7, 5, 9]

☐ [3, 7, 5, 9, 1]

☒ Compilation fails.

☐ An exception is thrown at runtime.

Question - 6

SCORE: 5 points

Java

Object Oriented Programming

Exception

Hard

Given that `Integer.parseInt()` throws `NumberFormatException`, and given:

```
3. public class Ladder {
4.     public static void main(String[] args) {
5.         try {
6.             System.out.println(doStuff(args));
7.         }
8.         catch (Exception e) {
9.             System.out.println("exc"); }
10.        doStuff(args);
11.    }
12.    static int doStuff(String[] args) {
13.        return Integer.parseInt(args[0]);
14.    }
```

And, if the code compiles, given the invocation:

`java Ladder x`

What is the result? (Choose all that apply.)

- ☐ 0
- ☐ exc
- ☒ "exc" followed by an uncaught exception.
- ☐ Compilation fails due to an error on line 4.
- ☐ Compilation fails due to an error on line 9.
- ☐ Compilation fails due to an error on line 11.
- ☐ An uncaught exception is thrown with no other output.