



F-code

Fcode Labs

TECHNICAL SUBMISSION

12/05/2022



Contents

Q1)..... 3

Q2)..... 4

Q3)..... 5

Q4)..... 5

Q1)

```
public class Question1 {
    public static void main(String[] args){
        //Since the question states "Given an array" I assumed that user
input is not necessary
        //Declaring a 2D array so all the integers could be added and cross-
checked and manually assigned
        int[][] numberArray=new int[6][2];
        numberArray[0][0]=3;
        numberArray[0][1]=4;
        numberArray[1][0]=1;
        numberArray[1][1]=2;
        numberArray[2][0]=5;
        numberArray[2][1]=2;
        numberArray[3][0]=7;
        numberArray[3][1]=10;
        numberArray[4][0]=4;
        numberArray[4][1]=3;
        numberArray[5][0]=2;
        numberArray[5][1]=5;
        //Two for loops are called for to traverse from so EVERY possible way
can be calculated
        //It will keep the left hand variable constant and check for all the
combinations on the right for example
        // [1,0],[1,1],[1,2] after the right hand reachess the counter it
would start again with 2, [2,0],[2,1],[2,2]
        for (int counter1 = 0; counter1 < 6; counter1++){ //This For loop is
for the first increment
            for (int counter2=0 ; counter2 < 6; counter2++){ //This For loop
is for the Second increment
                if (numberArray[counter1][0]==numberArray[counter2][1]){
                    //Compares the first value of the two integer array with
the second value of ALL the combinations
                    //and prints it out
                    System.out.println(numberArray[counter1][0]+"
"+numberArray[counter1][1]);
                }
            }
        }
    }
}
```

Q2)

```
import java.util.Scanner;
import java.util.HashSet;

public class Question2 {
    static boolean duplicate(int arr[], int k) {
        // Creates an empty hashset
        HashSet<Integer> kSet = new HashSet<>();
        for (int i = 0; i < arr.length; i++) {
            //if the array value is in the array return true
            if (kSet.contains(arr[i])) {
                return true;
            }
            // Adding it to the item set
            kSet.add(arr[i]);
            // when counter is greater than K the hashset value is removed
            // from the top as to keep the distance
            if (i >= k)
                kSet.remove(arr[i - k]);
        }
        return false;
    }

    public static void main(String[] args) {
        //Since the question says Given arrays , It is assumed the array can
        // be hardcoded
        Scanner userInput = new Scanner(System.in);
        int[] arrayOne = {5, 6, 8, 2, 4, 6, 9};
        int[] arrayTwo = {1, 2, 3, 2, 1};
        //Getting the K value from User
        System.out.println("Please enter your K value");
        int k = 0;
        Boolean valid = false;
        //puts the User in a loop until the K value is properly validated
        while (valid == false) { //valid will be true ONLY after K is verified
            //Try catch implemented to check if the input is a string or not
            try {
                k = Integer.parseInt(userInput.next());
                //if the K value extends the Array length then the array
                // length size assigned to K
                if (k > arrayOne.length) {
                    k = arrayOne.length;
                    valid = true;
                } else if (k > -1) { //if K is greater than 0 it is approved
                    valid = true;
                } else {
                    System.out.println("Please enter a positive number for
                    K");
                }
            }
        }
    }
}
```

```

        } catch (NumberFormatException e) {
            System.out.println("Please enter an Integer");
        }
    }
    //the array and K is fed to the fuction Duplicate and it returns
true IF a duplicate is found in the K distance
    if (duplicate(arrayOne, k)) {
        System.out.println("Duplicate found");
    } else {
        System.out.println("Duplicate not found");
    }
}
}

```

Q3)

- First I would make a variable called "highest" and assign it to 0
- Then I would make a for loop to traverse the whole array
- Then I would compare each element in the array to the highest
- If that element is higher than the variable highest then assign that element to Highest's
- Once the loop ends print out the highest

Q4)

I would use encapsulation to make sure data inside a certain object can't be accessed by other objects or a functions because I want to keep the data private and secure and only be accessed by an instance of that class. Abstraction would be used to hide certain functions or methods so that the user won't be able to see the code behind it and possibly exposing the code.

Polymorphism would be used to increase the code efficiency by making sure the same code is not repeated and the code and be re-used by other classes multiple classes would be able to use its functions. Inheritance can be used to decrease the code size and make the code efficient as things are not repeated again and again.