Group A

1. Given the following Java program:

```
import java.util.*;
public class Main
{
public static void main(String[] args) {
List num = new ArrayList(Arrays.asList(23, 16, 14, 33, 19, 6, 1));
System.out.println("List is "+num);
}
}
```

(a) Give the index values of all the odd numbers assuming zero-based indexing

| | |
|---|---|
| 23 | 0 |
| 33 | 3 |
| 19 | 4 |
| 1 | 6 |

(b) How many elements would be looked at when the list is traversed (from start to finish) until the value 19 was found?

=Until the 19 was found the no.of elements that would be looked when the list is traversed is 3.

2. Which of the following lists are syntactically correct in Java?

Try them out to see if you were correct.

(a) List num = new ArrayList(Arrays.asList(1, 2, 3, 'four'));
(b) List num = new ArrayList(Arrays.asList(1, 2, [3, 4]));

= List num = new ArrayList(Arrays.asList(1, 2, [3, 4])); are syntactically correct in java.

3. Perform a series of list operations on the following list:

List fruit = new ArrayList (Arrays.asList('apple', 'banana', 'pear', 'cherry'));

to produce this updated list:

['Grapefruit', 'banana', 'Date', 'cherry', 'Orange']

```
import java.util.*;
public class fruit_list
{
    public static void main(String[] args){
    List Fruit =new ArrayList (Arrays.asList("Apple", "Banana", "Pear", "Chreey"));
    System.out.println("The List of Fruits are:"+Fruit);
    Fruit.set(0,"Grapefruit");
    Fruit.set(2,"Date");
    Fruit.add(4,"Orange");
    System.out.println("The Updated List of Fruits are: "+Fruit);
    }
}
```

```
The List of Fruits are:[Apple, Banana, Pear, Chreey]
The Updated List of Fruits are: [Grapefruit, Banana, Date, Chreey, Orange]
```

Group B

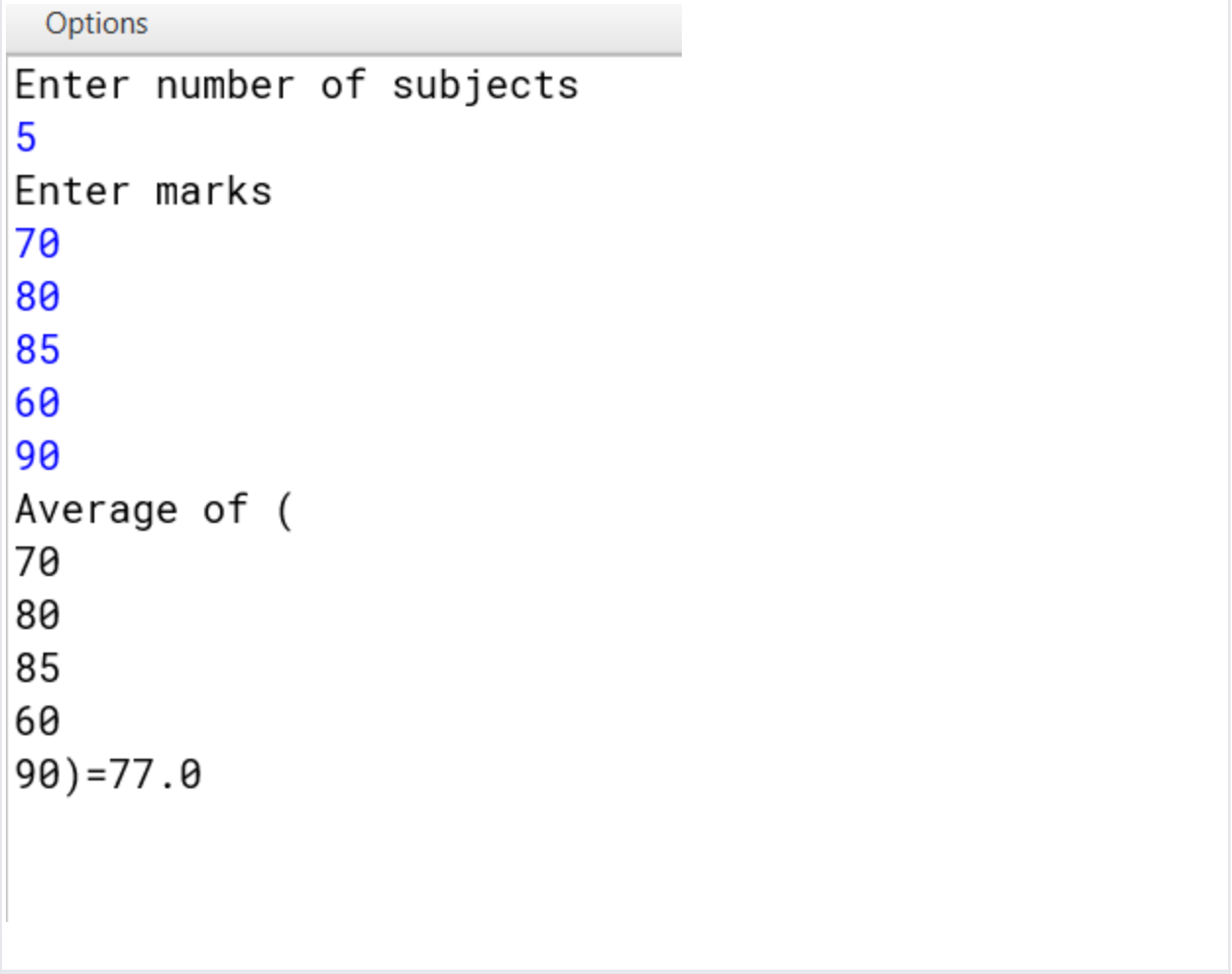
1. Write a program to find out whether a given integer is present in an array or not.

```java
public class inetegerpresentornot
{
    public static void main(String[] args) {

    int[] num = {1, 2, 3, 4, 5};
    int toFind = 8;
    boolean found = false;

    for (int n : num) {
      if (n == toFind) {
        found = true;
        break;
      }
    }

    if(found)
      System.out.println(toFind + " is found.");
    else
      System.out.println(toFind + " is not found.");
  }
}
```

```
8 is not found.
```

2. Calculate the average marks from an array containing marks of all students in physics using a for-each loop.

```java
import java.util.*;
public class averagearray
{
    public static void main(String[]args){
        int i;
        System.out.println("Enter number of subjects");
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int[] a=new int[n];
        double average=0;
        System.out.println("Enter marks");
        for(i=0;i<n;i++)
        {
            a[i]=sc.nextInt();
        }
        for(i=0;i<n;i++)
        {
            average=average+a[i];
        }
        System.out.println("Average of (");
        for(i=0;i<n-1;i++)
        {
            System.out.println(a[i]+"");
        }
        System.out.println(a[i]+")="+average/n);
        }

        }
```




3. Write a Java program to reverse an array.

```java
public class reversearray
{
    public static void main (String[]args){
        int[] array1=new int[]{10,20,30,40,50};
        System.out.println("The original array is:");
        for (int i=0;i<array1.length;i++){
            System.out.println(array1[i]);
        }
        System.out.println();
        System.out.println("The reversed array is :");
        for(int i=array1.length-1;i>=0;i--){
            System.out.println(array1[i]+"");

        }

        }

    }
```

Options

```
The original array is:
10
20
30
40
50

The reversed array is :
50
40
30
20
10
```

4. Write a Java program to find the maximum element in an array.

```java
public class maximumarray
{
    public static void main(String[]args){
        int arr[]={10,5,90,76,85};
        int maximum;
        {
            int i;
            int max=arr[0];
            for(i=1;i<arr.length;i++)
            if(arr[i]>max)
            max=arr[i];
            System.out.println("The maximum in given array is"+max);

        }

        }
    }
```

```
10
The maximum in given array is90
```

5. Write a Java program to find whether an array is sorted or not.

```java
public class arraysortedornot
{
    public static void main(String[]args){
        int array[]={1,2,3,4,8,6};
        System.out.println(isSorted(array));
     }
      private static boolean isSorted(int[]array){
          for(int i=0;i<array.length;i++)
              for(int j=i+1;j<array.length;j++)
                  if(array[j]<array[i])
                  return false;
                        return true;

                  }
                  }
```

```
false
```

Group C

1. Write a Java program to append the specified element to the end of a hash set.

```java
public class endhashset
{
    public static void main(String[] args) {
      // Create a empty hash set
      HashSet<String> h_set = new HashSet<String>();
    // use add() method to add values in the hash set
          h_set.add("Red");
          h_set.add("Green");
          h_set.add("Black");
          h_set.add("White");
          h_set.add("Pink");
          h_set.add("Yellow");

    // print the hash set
    System.out.println("The Hash Set: " + h_set);
    }
}
```

```
The Hash Set: [Red, White, Pink, Yellow, Black, Green]
```

2. Write a Java program to compare two sets and retain elements which are the same on both sets.

```java
import java.util.*;
public class sameonboth
{
    public static void main(String[] args){
        //Create a empty hashset
        HashSet<String> fruits1=new HashSet<String>();
        //use add() method to add values
        fruits1.add("Apple");
        fruits1.add("Guava");
        fruits1.add("Pomegranate");
        fruits1.add("Litchi");
        fruits1.add("Strawberry");
        //print the hash set
        System.out.println("First Hash set of fruits "+ fruits1);
        HashSet<String> fruits2=new HashSet<String>();
        //use add() method to add values
        fruits2.add("Apple");
        fruits2.add("Guava");
        fruits2.add("Pomegranate");
        fruits2.add("Litchi");
        fruits2.add("Strawberry");
        //print the hash set
        System.out.println("Second Hash set of fruits "+ fruits2);
        fruits1.retainAll(fruits2);
        System.out.println("Hash set of fruits:");
        System.out.println(fruits1);
    }
}
```

```
First Hash set of fruits [Guava, Apple, Litchi, Strawberry, Pomegranate]
Second Hash set of fruits [Guava, Apple, Litchi, Strawberry, Pomegranate]
Hash set of fruits:
[Guava, Apple, Litchi, Strawberry, Pomegranate]
```

3. Write a Java program to count the number of key-value mappings in a hash table

```java
import java.util.*;
public class keyvaluemaps
{
    public static void main(String []args){
    HashMap<Integer,String> hash_table=new HashMap<Integer,String>();
    hash_table.put(1, "Nepal");
    hash_table.put(2, "Australia");
    hash_table.put(3, "USA");
    hash_table.put(4, "London");
    hash_table.put(5, "UK");
    System.out.println("The size of the hash table: "+hash_table.size());
    }
}
```

```
The size of the hash table: 5
```

4. Write a Java program to get a collection view of the values contained in this map

```java
import java.util.*;
public class collectionviewmap
{
   public static void main(String args[]){

  HashMap<Integer,String> hash_map= new HashMap<Integer,String>();
    hash_map.put(1,"Pant");
  hash_map.put(2,"T-shirt");
  hash_map.put(3,"Jacket");
  hash_map.put(4,"Saree");
  hash_map.put(5,"Hoody");

   // checking collection view of the map
   System.out.println("Collection view is: "+ hash_map.values());
 }
}
```

```
Collection view is: [Pant, T-shirt, Jacket, Saree, Hoody]
```

(Optional)

Group D

1. Building a Rock Paper Scissor game in java

| |
|---|
| Ask the user to enter in their move.<br>Make a list of valid moves.<br>Check if the user entered a valid move by looking at the list of valid moves. (If the move is in the list, it is valid move)<br>Randomly generate the opponent's move. (Randomly choose one move from the list of valid moves)<br>Display the result to user<br>Use a loop to continue asking the user for their move.<br>Check if the user wants to quit. |