Report on s1709906's revision attempt at 2014's Inf1OP Programming Exam (sitting 1)

Generated by Automarker on Fri May 4 17:17:34 BST 2018

Question 1

Part 1a

Compiling Cost.java with the basic tests given to students in the exam succeeded.

Passed all 6 basic tests.

Compiling the submitted Cost.java with the test file worked fine.

Passed all 15 tests.

Marks for this part: 25 / 25

Submitted Cost.java

```

public class Cost {
 private int amount;
 private String currency;

 public Cost() {
  amount = 0;
  currency = "pounds␣sterling";
 }

 public int getAmount() {
  return amount;
 }

 public void setAmount(int newAmount) {
  if (newAmount >= 0) {
   this.amount = newAmount;
  }
 }

 public String getCurrency() {
  return currency;
 }

 public void setCurrency(String newCurrency) {
  boolean a = newCurrency.equals("pounds␣sterling");
  boolean b = newCurrency.equals("US␣dollars");
  boolean c = newCurrency.equals("euros");

  if (a || b || c ) {
   this.currency = newCurrency;
  }
 }

 public void convert (String newCurrency, double rate) {
  setCurrency(newCurrency);
  int newAmt = (int) Math.round(getAmount() / rate);
  setAmount(newAmt);
 }

 public String toString() {
  return getAmount() + "␣" + getCurrency();
 }

}
```

Part 1b

Compiling ActivityHoliday.java with the basic tests given to students in the exam succeeded.

Passed all 2 basic tests.

Compiling the submitted ActivityHoliday.java with the test file worked fine.

Passed all 9 tests.

Marks for this part: 25 / 25

Submitted ActivityHoliday.java

```
import java.util.HashMap;

class ActivityHoliday extends Holiday {

 private HashMap<String, Cost> activities = new HashMap<String, Cost>();

 public ActivityHoliday(String destination, int days) {
  super(destination, days);
  activities = new HashMap<String, Cost>(10);
 }

 public void addActivity(String activity, Cost cost) {
  activities.put(activity, cost);
 }

 public String toString() {
  String s = super.toString() + "\nActivities:";
  for (String key : activities.keySet()) {
   s = s + "\n" + key + "␣" + activities.get(key);
  }
  return s;
 }

 public static void main(String[] args) {
  ActivityHoliday a1 = new ActivityHoliday("Glasgow",14);
  Cost c1 = new Cost();
  c1.setAmount(5);
  c1.setCurrency("pounds␣sterling");
  a1.addActivity("Hunterian␣Museum", c1);
  a1.addActivity("Mackintosh␣House", c1);
  System.out.println(a1.toString());
 }
}
```

Marks for Question 1: 50/50

Question 2

Part 2a-e

Compiling Rearrangement.java with the basic tests given to students in the exam succeeded.

Passed all 5 basic tests.

Compiling the submitted Rearrangement.java with the test file worked fine.

Passed all 11 tests.

Marks for this part: 50 / 50

Submitted Rearrangement.java

```
import java.util.Arrays;

```

```java
public class Rearrangement {

 public static int dotWith(int[] a, int[] b) {
  if (a.length != b.length) {
   return 0;
  }

  int sum = 0;
  for (int i = 0; i < a.length; i++) {
   int product = a[i] * b[i];
   sum = sum + product;
  }

  return sum;
 }

 public static void rotate(int[] b) {
  int temp = b[b.length - 1];
  for (int i = b.length - 1 ; i > 0; i--) {
   b[i] = b[i - 1];
  }
  b[0] = temp;
 }

 public static int[] rotateStored(int[] b) {
  int[] result = new int[b.length];
  int temp = b[b.length - 1];
  for (int i = b.length - 1 ; i > 0; i--) {
   result[i] = b[i - 1];
  }
  result[0] = temp;

  return result;
 }

 public static int useRotations(int[] a, int[] b) {
  if (a.length != b.length) {
   return 0;
  }


  int[] rotated = new int[b.length];
  rotated = b;
  int max = dotWith(a,b);

  for (int i = 1; i < b.length; i++) {
   rotated = rotateStored(rotated);
   int dot = dotWith(a,rotated);
   if (dot > max) {
    max = dot;
   }
  }

  return max;
 }

 public static int useSorted(int[] a, int[] b) {

  if (a.length != b.length) {
   return 0;
  }

  Arrays.sort(a);
  Arrays.sort(b);

//  System.out.println(Arrays.toString(a));
//  System.out.println(Arrays.toString(b));

  return dotWith(a,b);


 }

 public static void main(String[] args) {
  int n = Integer.parseInt(args[0]);
  int[] a = new int[n];
  int[] b = new int[n];

  for (int i = 0; i < n; i++) {
   a[i] = Integer.parseInt(args[i + 1]);
  }

 // System.out.println(Arrays.toString(a));

  for (int i = 0; i < n; i++) {
   b[i] = Integer.parseInt(args[i + 1 + n]);
  }

 // System.out.println(Arrays.toString(b));

  System.out.println("dotWith␣gave:␣" + dotWith(a,b));
  System.out.println("useRotations␣gave:␣" + useRotations(a,b));
  System.out.println("useSorted␣gave:␣" + useSorted(a,b));
 }
}
```

Marks for Question 2: 50/50

Total marks: 100 / 100