



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**ΕΠΛ131 Αρχές Προγραμματισμού**

Ακαδημαϊκό Έτος 2016/17 – Εαρινό Εξάμηνο

**ΕΝΔΙΑΜΕΣΗ ΕΞΕΤΑΣΗ**

ΗΜΕΡΟΜΗΝΙΑ: 15 Μαρτίου 2017  
ΔΙΑΡΚΕΙΑ: 4:00μμ – 6:30μμ  
ΑΙΘΟΥΣΕΣ: Κτήριο ΧΩΔ01, Αίθουσα 108  
ΔΙΔΑΣΚΟΥΣΑ: Ελπίδα Κεραυνού-Παπαηλιού

**Απαντήστε και τις τρεις ερωτήσεις**  
Κάθε ερώτηση λαμβάνει 33 μονάδες.

### **Ερώτηση 1**

Έχουν αναπτυχθεί δύο εναλλακτικά προγράμματα (TwoNumsV1.java και TwoNumsV2.java) για την επίλυση του προβλήματος που περιγράφεται πιο κάτω.

Λαμβάνονται δύο φυσικοί αριθμοί, οι οποίοι περιλαμβάνουν αριθμητικά ψηφία πλην του 0. Το κάθε πρόγραμμα επεξεργάζεται τους δύο αριθμούς, και το αποτέλεσμα της επεξεργασίας παρουσιάζεται ως εξής:

- Αν οι αριθμοί έχουν διαφορετικό μήκος σε αριθμό ψηφίων, τότε απλά επισημαίνεται η διαφορά μήκους.
- Αν οι δύο αριθμοί έχουν το ίδιο μήκος, δεν είναι όμως οι ίδιοι αριθμοί αλλά περιλαμβάνουν ακριβώς τα ίδια ψηφία, τότε και πάλι επισημαίνεται αυτό.

- Αν οι δύο αριθμοί δεν είναι οι ίδιοι, αλλά έχουν το ίδιο μήκος, όμως δεν περιλαμβάνουν ακριβώς τα ίδια ψηφία, τότε παρουσιάζεται το κοινό τους μέρος (ίδια αντίστοιχα ψηφία) και τα διαφορετικά τους μέρη (διαφορετικά αντίστοιχα ψηφία).
- Αν φυσικά οι αριθμοί είναι οι ίδιοι, τότε απλά επισημαίνεται αυτό.

Για καλύτερη επεξήγηση της λειτουργίας των δύο προγραμμάτων, πιο κάτω δίνονται κάποια παραδείγματα χρήσης του προγράμματος TwoNumsV1 (οι ίδιες ακριβώς απαντήσεις θα δίνονταν στα συγκεκριμένα παραδείγματα και από το έτερο πρόγραμμα, TwoNumsV2):

```
$ java TwoNumsV1 567 12
```

```
Numbers 567 and 12 have different lengths!
```

```
$ java TwoNumsV1 45621 61542
```

```
Numbers 4562 and 6542 have the same digits but not in same positions!
```

```
$ java TwoNumsV1 1122234 5122678
```

```
Common digits of numbers 1122234 and 5122678 are: 122
```

```
Digits of number 1122234 that differ from number 5122678 are: 1234
```

```
Digits of number 5122678 that differ from number 1122234 are: 5678
```

```
$ java TwoNumsV1 785 785
```

```
Input numbers 785 and 785 are the same
```

Πιο κάτω δίνεται ο κώδικας των δύο προγραμμάτων, τα οποία είναι λογικά ισοδύναμα, αφού σε κάθε περίπτωση καταλήγουν στις ίδιες απαντήσεις. Για την απάντησή σας στην Ερώτηση 1 χρειάζεται να περιγράψετε αλγοριθμικά τη λύση του κάθε προγράμματος ξεχωριστά. Οι απαντήσεις σας πρέπει να είναι σε εννοιολογικό επίπεδο (conceptual level) και όχι απλά να συνιστούν μεταγραφή του κώδικα (code transliteration). Στη συνέχεια να κάνετε μια συγκριτική ανάλυση των εναλλακτικών αυτών λύσεων του ίδιου προβλήματος, με αναφορά στα ακόλουθα στοιχεία: κατανόηση/διαφάνεια, δομή/άρθρωση, επαναχρησιμοποίηση, δυνατότητα επεκτασιμότητας.

## Πρώτο Πρόγραμμα: TwoNumsV1.java

```
public class TwoNumsV1 {

    public static boolean sameLen (int x, int y){
        return Len(x) == Len(y);
    }

    public static int Len (int x){
        int l = 0;
        while (x!=0){l++; x = x/10;}
        return l;
    }

    public static int Sort (int x){
        return Sort(x/10, x%10);
    }

    public static int Sort (int x, int y){
        if (x == 0) return y;
        else return Sort(x/10, insert(y, x%10));
    }

    public static int insert (int y, int d){
        if (y == 0) return d;
        else if (d <= y%10) return y*10+d;
        else return insert(y/10, d)*10 + y%10;
    }

    public static int match (int x, int y){
        if (x==0 && y==0) return 0;
        else if (x%10 == y%10) return match(x/10, y/10)*10 + x%10;
        else return match (x/10, y/10);
    }

    public static int diff (int x, int y){
        if (x==0 && y==0) return 0;
        else if (x%10 != y%10) return diff(x/10, y/10)*10 + x%10;
        else return diff (x/10, y/10);
    }

    public static void main (String[] args){
        int x = Integer.parseInt(args[0]);
        int y = Integer.parseInt(args[1]);
        if (!sameLen(x,y))
            System.out.print("\n\nNumbers " + x + " and " + y +
                " have different lengths!");
        else if (x != y)
            if (Sort(x) == Sort(y))
                System.out.print("\n\nNumbers " + x + " and " + y +
                    " have the same digits but not in same positions!");
            else {int m = match(x,y); int dx = diff(x,y);
                int dy = diff(y,x);
                System.out.print("\n\nCommon digits of numbers " +
                    x + " and " + y + " are: " + m);
                System.out.print("\n\nDigits of number " + x +
                    " that differ from number " + y +
                    " are: " + dx);
                System.out.print("\n\nDigits of number " + y +
                    " that differ from number " + x +
                    " are: " + dy); }
    }
}
```

```

else System.out.print("\n\nInput numbers " + x + " and " + y +
    " are the same");

System.out.print("\n\n");
}

}

```

## Δεύτερο Πρόγραμμα: TwoNumsV2.java

```

public class TwoNumsV2 {

    public static void main (String[] args){

        int x = Integer.parseInt(args[0]);
        int y = Integer.parseInt(args[1]);
        int tx = x, ty = y, Lx = 0, Ly = 0;

        while (tx != 0){tx = tx/10; Lx++;}
        while (ty != 0){ty = ty/10; Ly++;}

        tx = x; ty = y;
        int m = 0, dx = 0, dy = 0;
        int i = Lx;

        if (Lx == Ly){
            while (tx != 0){
                int e = 1, j = i;
                while (j != 0) {e = e * 10; j--;}
                i--;
                int d1 = tx / e, d2 = ty / e;
                tx = tx % e; ty = ty % e;
                if (d1 == d2) m = m * 10 + d1;
                else {dx = dx * 10 + d1; dy = dy * 10 + d2;}
            }
        }

        int[] xs = {0,0,0,0,0,0,0,0,0,0};
        int[] ys = {0,0,0,0,0,0,0,0,0,0};
        boolean ok = Lx == Ly;
        tx = x; ty = y;

        if (ok){
            while (tx != 0){
                xs[tx%10]++; ys[ty%10]++;
                tx = tx/10; ty = ty/10;
            }
            for (int k = 1; k < 10; k++) if (xs[k] != ys[k]) ok = false;
        }

        if (Lx != Ly)
            System.out.print("\n\nNumbers " + x + " and " + y +
                " have different lengths!");
        else if (x != y)
            if (ok)
                System.out.print("\n\nNumbers " + x + " and " + y +
                    " have the same digits but not in same positions!");
            else {System.out.print("\n\nCommon digits of numbers " +
                x + " and " + y + " are: " + m);
                System.out.print("\n\nDigits of number " + x +

```

```

        " that differ from number " + y +
        " are: " + dx);
    System.out.print("\n\nDigits of number " + y +
        " that differ from number " + x +
        " are: " + dy); }
else System.out.print("\n\nInput numbers " + x + " and " + y +
    " are the same");

System.out.print("\n\n");
}

}

```

## Ερώτηση 2

Για τη δεύτερη ερώτηση χρειάζεται να λύσετε το ίδιο ουσιαστικά πρόβλημα με δύο εναλλακτικούς τρόπους όπως εξηγείται στη συνέχεια.

(α) Κατασκευάστε πρόγραμμα (Max2.java) το οποίο λαμβάνει από το χρήστη μια απροσδιορίστου μήκους ακολουθία ακεραίων αριθμών, η οποία τερματίζεται από τον αριθμό 0. Το 0 δεν ανήκει στην ακολουθία, απλά σηματοδοτεί το τέλος της. Στη συνέχεια το πρόγραμμα παρουσιάζει το μεγαλύτερο αριθμό της ακολουθίας, καθώς και το δεύτερο μεγαλύτερο αριθμό. Αν φυσικά ο χρήστης δεν δώσει κανένα αριθμό, ή αν όλοι οι αριθμοί που δώσει είναι οι ίδιοι, αυτό επισημαίνεται από το πρόγραμμα. **Το πρόγραμμα ΔΕΝ πρέπει να κάνει χρήση οποιουδήποτε πίνακα.**

Πιο κάτω δίνονται κάποια παραδείγματα χρήσης του προγράμματος Max2.

```
$ java Max2
```

```
Give number: 89
Give number: 5
Give number: 45
Give number: 89
Give number: 100
Give number: 0
```

```
The largest number is 100 and the second largest is 89
```

```
$ java Max2
```

```
Give number: 0
```

```
No input given
```

```
$ java Max2
```

```
Give number: 77
Give number: 77
Give number: 80
Give number: 80
Give number: 0
```

```
The largest number is 80 and the second largest is 77
```

```
$ java Max2
```

```
Give number: 6
```

```
Give number: 6
```

```
Give number: 6
```

```
Give number: 0
```

```
All numbers are the same >> 6
```

(β) Κατασκευάστε πρόγραμμα (MaxT.java) το οποίο λαμβάνει από τη γραμμή εντολής μια ακολουθία ακεραίων αριθμών (και πάλι απροσδιορίστου μήκους, αλλά τώρα δεν χρειάζεται να επισημανθεί το τέλος της όπως στο προηγούμενο πρόγραμμα) και όπως και το προηγούμενο πρόγραμμα παρουσιάζει το μεγαλύτερο αριθμό της ακολουθίας, καθώς και το δεύτερο μεγαλύτερο αριθμό. Επιπρόσθετα, και σε αντίθεση με το προηγούμενο πρόγραμμα, το δεύτερο πρόγραμμα δίνει και τις πρώτες σε σειρά θέσεις στις οποίες βρίσκονται οι δύο αριθμοί. Αν φυσικά δεν δοθούν καθόλου ορίσματα στη γραμμή εντολής, ή αν όλα τα ορίσματα είναι τα ίδια, όπως και προηγουμένως αυτό επισημαίνεται από το πρόγραμμα. **Το πρόγραμμα πρέπει να κάνει χρήση πίνακα.**

Πιο κάτω δίνονται τα αντίστοιχα παραδείγματα χρήσης για το πρόγραμμα MaxT.

```
$ java MaxT 89 5 45 89 100
```

```
The largest number is 100 found in position 5
```

```
and the second largest is 89 found in position 1
```

```
$ java MaxT
```

```
No Input given
```

```
$ java MaxT 77 77 80 80
```

```
The largest number is 80 found in position 3
```

```
and the second largest is 77 found in position 1
```

```
$ java MaxT 6 6 6
```

```
All numbers are the same >> 6
```

### Ερώτηση 3

Για την τρίτη ερώτηση χρειάζεται να συμπληρώσετε το πρόγραμμα Histogram.java, ο σκελετός του οποίου δίνεται πιο κάτω. Όπως λέει και το όνομά του (ιστόγραμμα) το πρόγραμμα αυτό δημιουργεί ιστογράμματα με διακριτό τρόπο και σε δύο διαφορετικές μορφές. Το πρόγραμμα λαμβάνει από τη γραμμή εντολής μια απροσδιορίστου μήκους ακολουθία ακεραίων αριθμών, και νοουμένου ότι δοθεί έστω ένας αριθμός, προχωρεί στο σχηματισμό των ιστογραμμάτων που αντιστοιχούν στους δεδομένους αριθμούς. Οι δύο διαφορετικές μορφές ιστογραμμάτων σχηματίζονται αντίστοιχα από τις μεθόδους, UHist (κάθετο ιστόγραμμα) και VHist (οριζόντιο ιστόγραμμα). Είναι αυτές τις δύο μεθόδους που χρειάζεται να ορίσετε, αξιοποιώντας όπου κρίνετε, βοηθητικές μεθόδους που ήδη ορίζονται (nl, line, max, min, positives, negatives) τις λειτουργίες των οποίων, όμως, χρειάζεται να εξηγήσετε. Η μέθοδος main ορίζεται επίσης.

```
public class Histogram {

    public static void nl() {System.out.println();}

    public static void line (int n, char c){
        for (int i = 1; i <= n; i++) System.out.print(c);
    }

    public static int max (int[] table){
        int res = table[0];
        for (int i = 1; i < table.length; i++)
            if (table[i] > res) res = table[i];
        return res;
    }

    public static int min (int[] table){
        int res = table[0];
        for (int i = 1; i < table.length; i++)
            if (table[i] < res) res = table[i];
        return res;
    }

    public static boolean positives (int [] table){
        for (int i = 0; i < table.length; i++)
            if (table[i] > 0) return true;

        return false;
    }

    public static boolean negatives (int [] table){
        for (int i = 0; i < table.length; i++)
            if (table[i] < 0) return true;

        return false;
    }
}
```

```

public static void UHist(int[] tab){
    . . . . .
}

public static void VHist(int[] table){
    . . . . .
}

public static void main(String[] args){

    if (args.length == 0)
        System.out.print("\n\nNo Data given\n\n");
    else {
        int [] table = new int[args.length];
        for (int i = 0; i < table.length; i++)
            table[i]=Integer.parseInt(args[i]);
        UHist(table);
        nl(); nl();
        VHist(table);
        nl(); nl();
    }
}
}

```

Οι λειτουργίες των μεθόδων UHist και VHist φαίνονται στα ακόλουθα παραδείγματα χρήσης του προγράμματος. Σε κάθε παράδειγμα, το πρώτο σχήμα αντιστοιχεί στη λειτουργία της UHist και το δεύτερο στη λειτουργία της VHist. Και οι δύο μέθοδοι επεξεργάζονται ακριβώς τα ίδια στοιχεία, δηλαδή τους αριθμούς που δίνονται στη γραμμή εντολής.

**UHist:** Οι στήλες που αντιστοιχούν στους αριθμούς, τυπώνονται κάθετα. Στην περίπτωση των θετικών αριθμών, οι στήλες έχουν κατεύθυνση προς τα πάνω και απαρτίζονται από τον χαρακτήρα \*. Στην περίπτωση κάποιου αριθμού που είναι το 0, δεν εμφανίζεται κανένα αστεράκι στη συγκεκριμένη στήλη. Στην περίπτωση των αρνητικών αριθμών οι στήλες έχουν κατεύθυνση προς τα κάτω και απαρτίζονται από το χαρακτήρα +. Οι πάνω και οι κάτω στήλες διαχωρίζονται μεταξύ τους από μια οριζόντια γραμμή που σχηματίζεται από το χαρακτήρα -.

**VHist:** Τα ιστογράμματα σχηματίζονται σε οριζόντια τοποθέτηση, όπου οι σειρές που αντιστοιχούν στους θετικούς αριθμούς προσανατολίζονται προς τα δεξιά και απαρτίζονται από το χαρακτήρα \*, ενώ οι σειρές που αντιστοιχούν στους αρνητικούς αριθμούς προσανατολίζονται προς τα αριστερά και απαρτίζονται από το χαρακτήρα +. Ο διαχωρισμός ανάμεσα στα θετικά και τα



αρνητικά, τώρα γίνεται με μια κάθετη γραμμή, η οποία απαρτίζεται από το χαρακτήρα |.

Και οι δύο μέθοδοι έχουν ως παράμετρο ένα πίνακα ακεραίων. Για λόγους που πρέπει να εξηγήσετε οι πίνακες που λαμβάνουν ως ορίσματα οι εν λόγω μέθοδοι δεν θα πρέπει να τροποποιούνται. Φυσικά αν η επεξεργασία που χρειάζεται να γίνει συνεπάγεται τροποποίηση του πίνακα, ο πίνακας μπορεί να αντιγραφεί σε τοπικό πίνακα και η επεξεργασία να γίνεται στον τοπικό πίνακα.

```
$ java Histogram 7 10 -8 -9 15 0 2 6 0 0 -10 -5 13 6
```

```

      *
      *
      *      *
      *      *
      *      *
 *   *      *
 *   *      *
 *   *      *
**  *      *
**  * *   **
**  * *   **
**  * *   **
**  * *   **
**  * **  **
**  * **  **
-----
  ++      ++
  ++      ++
  ++      ++
  ++      ++
  ++      +
  ++      +
  ++      +
  +       +
         +

          | *****
          | *****
+++++++ |
+++++++ |
          | *****
          |
          | **
          | *****
          |
          |
+++++++ |
+++++++ |
          | *****
          | *****

```

```
$ java Histogram 1 2 3 4 5 4 3 2 1 -1 -2 -3 -4 -5 -4 -3 -2 -1
```

```
 *
 ***
 *****
 *********
 *****
 -----
          ++++++++
         ++++++++
        ++++++
       +++++
      +++
     +
```

```
 | *
 | **
 | ***
 | ****
 | *****
 | ****
 | ***
 | **
 | *
+ |
++ |
+++ |
++++ |
+++++ |
+++++ |
+++++ |
+++ |
++ |
+ |
```

**ΤΕΛΟΣ ΕΡΩΤΗΣΕΩΝ**

## ΧΩΡΟΣ ΑΠΑΝΤΗΣΕΩΝ

Όνοματεπώνυμο Φοιτητή: -----

Ταυτότητα: -----

Υπογραφή: -----

| <b>Ερωτήσεις</b>      | <b>Μονάδες</b> |
|-----------------------|----------------|
| Ερώτηση 1             |                |
| Ερώτηση 2             |                |
| Ερώτηση 3             |                |
| <b>Σύνολο Μονάδων</b> |                |

Παρατηρήσεις Διδάσκοντα

## Απάντηση στην Ερώτηση 1

**(τυχόν συνέχεια στην απάντηση της Ερώτησης 1)**

**(τυχόν συνέχεια στην απάντηση της Ερώτησης 1)**

## **Απάντηση στην Ερώτηση 2**

**(τυχόν συνέχεια στην απάντηση της Ερώτησης 2)**



**(τυχόν συνέχεια στην απάντηση της Ερώτησης 2)**

### **Απάντηση στην Ερώτηση 3**

**(τυχόν συνέχεια στην απάντηση της Ερώτησης 3)**

**(τυχόν συνέχεια στην απάντηση της Ερώτησης 3)**

**Επιπρόσθετο φύλλο (για συμπλήρωση απάντησης ή πρόχειρο)**

**Επιπρόσθετο φύλλο (για συμπλήρωση απάντησης ή πρόχειρο)**

**Επιπρόσθετο φύλλο (για συμπλήρωση απάντησης ή πρόχειρο)**

**Επιπρόσθετο φύλλο (για συμπλήρωση απάντησης ή πρόχειρο)**

**ΤΕΛΟΣ ΕΞΕΤΑΣΤΙΚΟΥ ΔΟΚΙΜΙΟΥ**