



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**ΕΠΛ131 Αρχές Προγραμματισμού**

Ακαδημαϊκό Έτος 2016/17 – Χειμερινό Εξάμηνο

**ΤΕΛΙΚΗ ΕΞΕΤΑΣΗ ΕΞΑΜΗΝΟΥ**

ΗΜΕΡΟΜΗΝΙΑ: 12 Δεκεμβρίου 2016  
ΔΙΑΡΚΕΙΑ: 8:30πμ – 11:00πμ  
ΑΙΘΟΥΣΕΣ: Κτήριο ΧΩΔ01, Αίθουσες 108 και 109  
ΔΙΔΑΣΚΟΥΣΑ: Ελπίδα Κεραυνού-Παπαηλιού

**Απαντήστε όλες τις ερωτήσεις**

Κάθε ερώτηση βαθμολογείται με 25 μονάδες.

Καλή επιτυχία!

### **Ερώτηση 1**

Σας δίνεται η κλάση αντικειμένου Node ως ακολούθως:

```
public class Node {
    private int elem;
    private int len;
    private int occurs;

    public Node(int e, int m, int o){
        elem = e; len = m; occurs = o;
    }

    public void incLen(){len++;}
    public void incOccur(){occurs++;}

    public int getElem(){return elem;}
    public int getLen(){return len;}
    public int getOccurs(){return occurs;}

    public String toString(){
        String s = "Subsequence of length " + len +
            " consisting of number " + elem +
            " occurs " + occurs + " times";
        return s;
    }
}
```

Κάθε περίπτωση αντικειμένου Node αποτελείται από τρία πεδία, elem, len και occurs, που αντίστοιχα αντιπροσωπεύουν μια (υπο)ακολουθία μήκους len στοιχείων αποτελούμενη από το στοιχείο elem, η οποία εμφανίζεται occurs φορές σε κάποια συμφραζόμενα, π.χ. σε κάποια άλλη ακολουθία.

Το πρόγραμμα MaxSeq.java είναι client της Node. Το πρόγραμμα αυτό λαμβάνει από τη γραμμή εντολής μια ακολουθία φυσικών αριθμών, οποιουδήποτε μήκους, την οποία αναλύει στις υπο-ακολουθίες ιδίων στοιχείων που περιλαμβάνει, όπου κάθε υπο-ακολουθία προσδιορίζεται από το επαναλαμβανόμενο στοιχείο της, το μήκος της, και το πλήθος των εμφανίσεων της στη δεδομένη ακολουθία αριθμών. Οι εν λόγω υπο-ακολουθίες παρουσιάζονται, και επίσης υποδεικνύεται η πρώτη σε σειρά και μέγιστου μήκους υπο-ακολουθία. Η λειτουργία του προγράμματος MaxSeq διαφαίνεται στα ακόλουθα παραδείγματα χρήσης του:

```
$ java MaxSeq 1 1 1 0 0 1 1 1 0 0 1 1 1 0 0 1 1 1
```

The subsequences are:

```
Subsequence of length 3 consisting of number 1 occurs 4 times  
Subsequence of length 2 consisting of number 0 occurs 3 times
```

Thus the first subsequence of max length is:

```
Subsequence of length 3 consisting of number 1 occurs 4 times
```

```
$ java MaxSeq 1 2 2 5 5 5 1 2 9 9 9 3 3 3
```

The subsequences are:

```
Subsequence of length 1 consisting of number 1 occurs 2 times  
Subsequence of length 2 consisting of number 2 occurs 1 times  
Subsequence of length 3 consisting of number 5 occurs 1 times  
Subsequence of length 1 consisting of number 2 occurs 1 times  
Subsequence of length 3 consisting of number 9 occurs 1 times  
Subsequence of length 3 consisting of number 3 occurs 1 times
```

Thus the first subsequence of max length is:

```
Subsequence of length 3 consisting of number 5 occurs 1 times
```

```
$ java MaxSeq 9 9 9 9 9 9
```

The subsequences are:

```
Subsequence of length 6 consisting of number 9 occurs 1 times
```

Thus the first subsequence of max length is:

```
Subsequence of length 6 consisting of number 9 occurs 1 times
```

```
$ java MaxSeq 1 1 1 2 2 2 2 3 3 3 3 3
```

The subsequences are:

```
Subsequence of length 3 consisting of number 1 occurs 1 times  
Subsequence of length 4 consisting of number 2 occurs 1 times  
Subsequence of length 5 consisting of number 3 occurs 1 times
```

Thus the first subsequence of max length is:  
Subsequence of length 5 consisting of number 3 occurs 1 times

Πιο κάτω σας δίνετε ο σκελετός του προγράμματος MaxSeq, τον οποίο χρειάζεται να συμπληρώσετε, ορίζοντας τις μεθόδους insertNode, maxNode, display και maxS, των οποίων οι λειτουργίες εξηγούνται στα αντίστοιχα σχόλια πιο κάτω. Σημειώνεται ότι η συνάρτηση main σας δίνετε ήδη ορισμένη.

```
public class MaxSeq {

    public static int insertNode(Node[]ns, Node n, int count){
        /* Θεωρώντας ότι μόνο οι πρώτες count είσοδοι του πίνακα ns
           περιλαμβάνουν έγκυρα στοιχεία, προστίθεται ο n στον ns,
           νοουμένου ότι ο ns δεν περιλαμβάνει ήδη τη σχετική ακολουθία
           αριθμών. Διαφορετικά αν ήδη την περιλαμβάνει, τότε απλά
           αυξάνει κατά ένα τον αριθμό εμφανίσεων της. Η μέθοδος
           επιστρέφει το (ενδεχομένως αναθεωρημένο)πλήθος των στοιχείων
           του ns μετά την επεξεργασία του n. */

        . . . . .

    }

    public static void display(Node[] ns, int count){

        /* Θεωρώντας ότι μόνο οι πρώτες count είσοδοι του πίνακα ns
           περιλαμβάνουν έγκυρα στοιχεία, παρουσιάζει τα εν λόγω στοιχεία
           όπως διαφαίνεται στα πιο πάνω παραδείγματα χρήσης του
           προγράμματος */

        . . . . .

    }

    public static Node maxNode(Node[]ns, int count){
        /* Θεωρώντας ότι μόνο οι πρώτες count είσοδοι του πίνακα ns
           περιλαμβάνουν έγκυρα στοιχεία, και αφού πρώτα παρουσιάσει
           με τη χρήση της display τα έγκυρα στοιχεία του ns, επιστρέφει
           το πρώτο σε σειρά στοιχείο του ns, που περιλαμβάνει μέγιστη σε
           μήκος ακολουθία αριθμών. */

        . . . . .

    }

    public static Node maxS (int[] table){

        /* Αξιοποιώντας την insertNode, και κατ' επέκταση την κλάση
           αντικειμένου Node, επεξεργάζεται την ακολουθία αριθμών που
           βρίσκεται στον πίνακα table για να δημιουργήσει τις σχετικές
           υπο-ακολουθίες αριθμών. Στη συνέχεια μέσω της maxNode,
           παρουσιάζει τις εν λόγω υπο-ακολουθίες και επίσης επιστρέφει
           την πρώτη σε σειρά και μέγιστου μήκους υπο-ακολουθία. */

        . . . . .

    }

}
```

```

public static void main(String[] args){

    int[] table = new int[args.length];
    for (int i = 0; i < args.length; i++)
        table[i] = Integer.parseInt(args[i]);

    Node n = maxS(table);

    System.out.print(
        "\n\nThus the first subsequence of max length is:\n" + n + "\n\n");

}
}

```

## Ερώτηση 2

Αναπτύξτε το πρόγραμμα Fstatistics.java για την παρουσίαση στοιχείων αναφορικά με μια συγκεκριμένη αεροπορική πτήση, η οποία εκτελείται σε καθημερινή βάση. Έστω ότι το αρχείο flights.txt περιλαμβάνει τα σχετικά στοιχεία ως ακολούθως:

112	34	16
25	34	95
67	45	30
96	100	0
100	100	50
65	89	32
72	0	6
56	60	20
31	50	30
67	41	23

Οι γραμμές του αρχείου παρουσιάζουν τα στοιχεία για τη δεδομένη πτήση σε διαδοχικές ημέρες. Οι τρεις αριθμοί αντιπροσωπεύουν πόσοι άνδρες, γυναίκες και παιδιά ταξίδεψαν στην εν λόγω πτήση τη συγκεκριμένη ημέρα, π.χ. την πρώτη μέρα ταξίδεψαν 112 άνδρες, 34 γυναίκες και 16 παιδιά, ενώ τη δέκατη μέρα, που τυγχάνει να είναι και η τελευταία μέρα για την οποία περιλαμβάνει στοιχεία το αρχείο flights.txt, ταξίδεψαν 67 άνδρες, 41 γυναίκες και 23 παιδιά.

Για την παρουσίαση των στοιχείων δημιουργείται ένα ιστόγραμμα στο οποίο για κάθε μέρα παρουσιάζεται ο αριθμός ανδρών, ο αριθμός γυναικών, ο αριθμός παιδιών, και το σύνολο αυτών (όλοι αυτοί οι αριθμοί είναι στρογγυλεμένοι στην πλησιέστερη δεκάδα, και ο κάθε χαρακτήρας αντιπροσωπεύει μια δεκάδα). Παρουσιάζονται επίσης οι πρώτες σε σειρά μέρες κατά τις οποίες ταξίδεψε ο μέγιστος αριθμός ανδρών, γυναικών, παιδιών και συνόλου επιβατών, αντίστοιχα. Σε κάθε περίπτωση αναφέρεται ο συγκεκριμένος μέγιστος αριθμός κατηγορίας/συνόλου επιβατών καθώς επίσης και η συγκεκριμένη μέρα στην οποία αναφέρεται ο αριθμός αυτός.

Πιο κάτω δίνεται το παράδειγμα εκτέλεσης του προγράμματος με input (μέσω file redirection) από το πιο πάνω αρχείο:

```
$ java Fstatistics < flights.txt
```

```
1 MMMMMMMMMMM
1 WWW
1 CC
1 TTTTTTTTTTTTTTTT
2 MMM
2 WWW
2 CCCCCCCCCC
2 TTTTTTTTTTTTTTTT
3 MMMMMMM
3 WWWWW
3 CCC
3 TTTTTTTTTTTTTTTT
4 MMMMMMMMMMM
4 WWWWWWWWWWW
4
4 TTTTTTTTTTTTTTTTTT
5 MMMMMMMMMMM
5 WWWWWWWWWWW
5 CCCCC
5 TTTTTTTTTTTTTTTTTT
6 MMMMMMM
6 WWWWWWWWWWW
6 CCC
6 TTTTTTTTTTTTTTTTTT
7 MMMMMMM
7
7 C
7 TTTTTTTT
8 MMMMMM
8 WWWWWW
8 CC
8 TTTTTTTTTTTTTTTT
9 MMM
9 WWWWW
9 CCC
9 TTTTTTTTTTTT
10 MMMMMMM
10 WWWW
10 CC
10 TTTTTTTTTTTTTTTT
```

```
Max no of men was 112 on day 1
Max no of women was 100 on day 4
Max no of children was 95 on day 2
Max no of total passengers was 250 on day 5
```

### Ερώτηση 3

(α) Αναπτύξτε το πρόγραμμα Caps.java, το οποίο διαβάζει ένα κείμενο και το εκτυπώνει, όπου τα μικρά γράμματα αντικαθίστανται με κεφαλαία γράμματα. Ακολουθεί παράδειγμα χρήσης του προγράμματος, όπου η επεξεργασία γίνεται στο αρχείο Node.java (μέσω file redirection), το περιεχόμενο του οποίου δίνεται στην αρχή της Ερώτησης 1:

```
$ java Caps < Node.java
```

```

PUBLIC CLASS NODE {

    PRIVATE INT ELEM;
    PRIVATE INT LEN;
    PRIVATE INT OCCURS;

    PUBLIC NODE (INT E, INT M, INT O) {
        ELEM = E; LEN = M; OCCURS = O;
    }

    PUBLIC VOID INCLLEN() {LEN++;}
    PUBLIC VOID INCOCCUR() {OCCURS++;}

    PUBLIC INT GETELEM() {RETURN ELEM;}
    PUBLIC INT GETLEN() {RETURN LEN;}
    PUBLIC INT GETOCCURS() {RETURN OCCURS;}

    PUBLIC STRING TOSTRING() {
        STRING S = "SUBSEQUENCE OF LENGTH " + LEN +
            " CONSISTING OF NUMBER " + ELEM +
            " OCCURS " + OCCURS + " TIMES";

        RETURN S;
    }
}

```

(β) Αναπτύξτε το πρόγραμμα LineN.java το οποίο διαβάζει ένα κείμενο και το εκτυπώνει με αριθμημένες τις γραμμές του. Επισημαίνεται ότι οι γραμμές ενός κειμένου διαχωρίζονται μεταξύ τους από τον ειδικό χαρακτήρα '\n'. Ακολουθεί παράδειγμα χρήσης του προγράμματος, όπου η επεξεργασία και πάλι γίνεται στο αρχείο Node.java, και πάλι μέσω file redirection:

```

$ java LineN < Node.java
1: public class Node {
2:
3:     private int elem;
4:     private int len;
5:     private int occurs;
6:
7:     public Node(int e, int m, int o){
8:         elem = e; len = m; occurs = o;
9:     }
10:
11:     public void incLen(){len++;}
12:     public void incOccur(){occurs++;}
13:
14:     public int getElem(){return elem;}
15:     public int getLen(){return len;}
16:     public int getOccurs(){return occurs;}
17:
18:     public String toString(){
19:         String s = "Subsequence of length " + len +
20:             " consisting of number " + elem +
21:             " occurs " + occurs + " times";
22:         return s;
23:     }
24:
25: }

```

(γ) Αναπτύξτε το πρόγραμμα WCount.java το οποίο διαβάζει ένα κείμενο και εκτυπώνει τρεις φυσικούς αριθμούς, έστω lines, words και chars, όπου lines είναι ο αριθμός των γραμμών, words είναι ο αριθμός των λέξεων, και chars είναι ο αριθμός των χαρακτήρων, που περιλαμβάνονται στο συγκεκριμένο κείμενο. Όπως ήδη αναφέρθηκε οι γραμμές ενός κειμένου διαχωρίζονται μεταξύ τους από τον ειδικό χαρακτήρα '\n', ενώ οι λέξεις διαχωρίζονται μεταξύ τους από τους λεγόμενους χαρακτήρες "white space" δηλαδή το space (' '), το new line ('\n'), και το tab ('\t'), όπως ορίζεται στην ακόλουθη βοηθητική συνάρτηση white\_space:

```
public static boolean white_space (char c) {
    return c == ' ' || c == '\n' || c == '\t';
}
```

Ακολουθεί παράδειγμα χρήσης του προγράμματος, όπου η επεξεργασία και πάλι γίνεται στο αρχείο Node.java, και πάλι μέσω file redirection:

```
$ java WCount < Node.java
chars = 537, words = 81, lines = 25
```

## Ερώτηση 4

Πιο κάτω σας δίνετε μέρος του προγράμματος XmasFigs.java το οποίο δημιουργεί διακριτά σχήματα Χριστουγεννιάτικου περιεχομένου:

```
public class XmasFigs {

public static void snowManRow (int row){
    switch (row){
        case 1: System.out.printf("                "); break;
        case 2: System.out.printf("      /_____\\  "); break;
        case 3: System.out.printf("      | x   x |   "); break;
        case 4: System.out.printf("      |   o   |   "); break;
        case 5: System.out.printf("  _|   \\ \\  --- /   |_  "); break;
        case 6: System.out.printf("  \\ \\  ----- /_  "); break;
        case 7: System.out.printf("    \\ \\ /           \\ \\ /  "); break;
        case 8: System.out.printf("      |   o   |   "); break;
        case 9: System.out.printf("      |   o   |   "); break;
        case 10: System.out.printf("      |   o   |   "); break;
        case 11: System.out.printf("      |           |   "); break;
        case 12: System.out.printf("      \\ \\_____ /    "); break;
        case 13: System.out.printf(" * MERRY XMAS * ");
    }
}

public static void starRow (int row){
    switch (row){
        case 1: System.out.print("      *      "); break;
        case 2: System.out.print("     ***     "); break;
        case 3: System.out.print(" ***** "); break;
        case 4: System.out.print(" ***** "); break;
        case 5: System.out.print(" ***** "); break;
        case 6: System.out.print("     ***     "); break;
        case 7: System.out.print("      *      ");
    }
}
```

```

public static void xmasRow (int row){
    switch (row){
        case 1: System.out.print("          "); break;
        case 2: System.out.print("          "); break;
        case 3: System.out.print("    MERRY  "); break;
        case 4: System.out.print("  X M A S  "); break;
        case 5: System.out.print(" *****  "); break;
        case 6: System.out.print("          "); break;
        case 7: System.out.print("          "); break;
    }
}

public static void blankRow(){
    System.out.print("          ");
}

. . . . .

public static void main(String[] args){

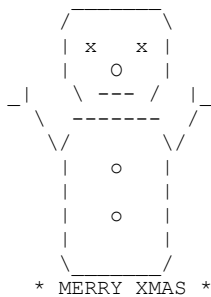
    . . . . .

}
}

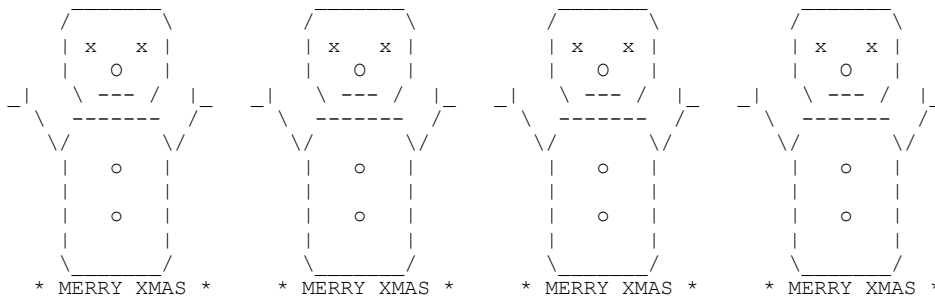
```

Βάσει επιλογής του χρήστη η οποία δίνεται στη γραμμή εντολής, το πρόγραμμα σχεδιάζει είτε μια σειρά δεδομένου μεγέθους από χιονάνθρωπους, είτε ένα μοτίβο (που και πάλι προσδιορίζει ο χρήστης) από αστέρια, όπως δεικνύεται στα ακόλουθα παραδείγματα, και εξηγείται πιο αναλυτικά στη συνέχεια.

```
$ java XmasFigs 1 1
```



```
$ java XmasFigs 1 4
```







```
$ java XmasFigs 2 < pattern3.txt
```

```

*
***
*****
*****
*****
***
*

          *
          ***
          *****
          *****
          *****
          ***
          *

          *
          ***
          *****
          *****
          *****
          ***
          *

MERRY
XMAS
*****

          *
          ***
          *****
          *****
          *****
          ***
          *

          *
          ***
          *****
          *****
          *****
          ***
          *

*
***
*****
*****
*****
***
*

          *
          ***
          *****
          *****
          *****
          ***
          *

```

Όπως αναφέρθηκε, και όπως διαφαίνεται στα πιο πάνω παραδείγματα χρήσης του προγράμματος, υπάρχουν δύο επιλογές, οι οποίες προσδιορίζονται με τον αριθμό 1 ή τον αριθμό 2, ως το πρώτο όρισμα στη γραμμή εντολής.

Η επιλογή νούμερο 1 δημιουργεί μια σειρά από χιονάνθρωπους. Το πλήθος των χιονάνθρωπων δίνεται ως το δεύτερο όρισμα στη γραμμή εντολής. Στα πιο πάνω παραδείγματα αυτής της επιλογής, αντίστοιχα δημιουργείται μια σειρά από ένα χιονάνθρωπο και μια σειρά από τέσσερις χιονάνθρωπους. Για την υλοποίηση αυτής της επιλογής αξιοποιείστε τη συνάρτηση `snowManRow(.)` που σας δίνετε.

Η επιλογή νούμερο 2 δημιουργεί ένα μοτίβο από αστέρια. Το μοτίβο δίνεται σε ένα αρχείο κειμένου το οποίο προσδιορίζεται μέσω `file redirection` όπως δεικνύεται στα πιο πάνω τρία παραδείγματα αυτής της επιλογής. Στα εν λόγω παραδείγματα τα περιεχόμενα των δεδομένων αρχείων κειμένου είναι τα ακόλουθα:

Αρχείο `pattern1.txt`

```
3
false true false
true false true
false true false
```

Αρχείο pattern2.txt

```
5
false false true  false false
false true  false true  false
true  false false false true
false true  false true  false
false false true  false false
```

Αρχείο pattern3.txt

```
5
true  false false false true
false true  false true  false
false false true  false false
false true  false true  false
true  false false false true
```

Η δομή των πιο πάνω αρχείων είναι η ακόλουθη: Πρώτα δίνεται το μέγεθος του τετραγωνικού πλέγματος (3x3, 5x5 και 5x5 στα πιο πάνω παραδείγματα). Στη συνέχεια προσδιορίζονται τα τμήματα του πλέγματος που θα περιέχουν αστέρια (εκεί που υπάρχει η τιμή true). Στο μέσο του πλέγματος (ασχέτως ποια λογική τιμή αναφέρεται) πάντα μπαίνει το σχετικό μήνυμα, όπως φαίνεται και στα πιο πάνω παραδείγματα. Για την υλοποίηση αυτής της επιλογής αξιοποιείστε τις συναρτήσεις `starRow(.)`, `xmasRow(.)` και `blankRow()` που σας έχουν δοθεί.

Ολοκληρώστε το πρόγραμμα με την υλοποίηση των πιο πάνω επιλογών.

Τέλος, για ένα **“bonus”** 12 μονάδων, αναπτύξτε το βοηθητικό πρόγραμμα `CreatePattern.java` για τη δημιουργία μοτίβων δεδομένου μεγέθους της μορφής που δεικνύεται στο τελευταίο παράδειγμα πιο πάνω, δηλαδή όπως το `pattern3.txt`, βάσει των οποίων τα αστέρια τοποθετούνται στις διαγώνιους. Για παράδειγμα,

```
$ java CreatePattern 7 > pat7.txt
```

δημιουργεί το αρχείο `pat7.txt` με το ακόλουθο περιεχόμενο:

```
7
true  false false false false false true
false true  false false false true  false
false false true  false true  false false
false false false true  false false false
false false true  false true  false false
false true  false false false true  false
true  false false false false false true
```

και επομένως η εντολή

```
$ java CreatePattern 3 | java XmasFigs 2
```

δημιουργεί το σχήμα που ακολουθεί:

\*  
\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*  
\*

\*  
\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*  
\*

MERRY  
X M A S  
\*\*\*\*\*

\*  
\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*  
\*

\*  
\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*  
\*

**ΤΕΛΟΣ ΕΡΩΤΗΣΕΩΝ**

## ΧΩΡΟΣ ΑΠΑΝΤΗΣΕΩΝ

Όνοματεπώνυμο Φοιτητή: -----

Ταυτότητα: -----

Υπογραφή: -----

<b>Ερωτήσεις</b>	<b>Μονάδες</b>
Ερώτηση 1	
Ερώτηση 2	
Ερώτηση 3	
Ερώτηση 4	
<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)**

## Απάντηση στην Ερώτηση 4

**(τυχόν συνέχεια στην απάντηση της Ερώτησης 4)**



**(τυχόν συνέχεια στην απάντηση της Ερώτησης 4)**

**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)**

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

**ΤΕΛΟΣ ΕΞΕΤΑΣΤΙΚΟΥ ΔΟΚΙΜΙΟΥ**