Δομημένος Προγραμματισμός – Εργαστήριο 11
Δείκτες - Pointers

Άσκηση 11.1

Να γραφεί Πρόγραμμα `C`, το οποίο **<u>χωρίς τη χρήση αγκυλών</u>**, ούτε στη `main()` ούτε στις συναρτήσεις, αλλά με τη χρήση **δεικτών – pointers** και τη χρήση της **μνήμης σωρού** με την εντολή `malloc()` θα κάνει τα παρακάτω :

- Θα **διαβάζει** μια τιμή για το `n > 5`.
- Θα **γεμίζει** έναν πίνακα ακεραίων `n` θέσεων με τυχαίες τιμές στο `[1,10]` καλώντας τη συνάρτηση `fillPin()`.
- Θα **εμφανίζει** τα περιεχόμενα του πίνακα καλώντας τη συνάρτηση `showPin()`.
- Θα **βρίσκει** τη θέση του στοιχείου με τη μεγαλύτερη τιμή και του στοιχείου με τη μικρότερη τιμή καλώντας τις συναρτήσεις `findThesiMax()` και `findThesiMin()` αντίστοιχα. Οι δύο συναρτήσεις θα **επιστρέφουν** τη θέση του **μεγίστου** και του **ελαχίστου** στοιχείου αντίστοιχα. Στη `main()`, θα **εμφανίζει** το μέγιστο και το ελάχιστο στοιχείο καθώς και τις αντίστοιχες θέσεις τους στον πίνακα.
- Θα **βρίσκει** πόσα στοιχεία είναι πάνω από τον μέσο όρο των στοιχείων του πίνακα καλώντας την συνάρτηση `countAvg()`. Η συνάρτηση θα **επιστρέφει** το πλήθος των στοιχείων που είναι πάνω από των μέσο όρο. Το πλήθος θα εμφανίζεται στην `main()`.
- Θα **διαβάζει** μια ακέραια τιμή για το `num`.
- Θα **καλεί** την συνάρτηση `searchPThesiNum`() με την οποία θα βρίσκει τη θέση του `num` στον πίνακα, αν υπάρχει.
- Θα **εμφανίζει** τη θέση που βρέθηκε το `num` και το στοιχείο του πίνακα, αν βρέθηκε, διαφορετικά, το μήνυμα “NOT FOUND”.
- Θα **καλεί** την συνάρτηση `bubbleDesc`() με την οποία θα ταξινομεί τα στοιχεία του πίνακα με την μέθοδο bubble sort, αλλά σε φθίνουσα τάξη.
- Θα **εμφανίζει** τα περιεχόμενα του πίνακα καλώντας τη μέθοδο `showPin()`.
- Θα **διαβάζει** μια ακέραια τιμή για το `num`.
- Θα **καλεί** την συνάρτηση `binarysearchPThesiNum`() με την οποία θα βρίσκει τη θέση του `num` στον πίνακα, αν υπάρχει. Οι τιμές της αρχής και του τέλους να περνάνε παραμετρικά.
- Θα **εμφανίζει** τη θέση που βρέθηκε το `num` και το στοιχείο του πίνακα, αν βρέθηκε, διαφορετικά, το μήνυμα “NOT FOUND”.
- Θα **καλεί** την συνάρτηση `binarysearchPThesiNumRec`(), η οποία βρίσκει με δυαδική αναζήτηση και επιστρέφει τη θέση του `num` στον ταξινομημένο πίνακα, αν υπάρχει, καλώντας τον εαυτό της αναδρομικά χωρίς τη χρήση δεικτών.
- Θα **εμφανίζει** τη θέση που βρέθηκε το `num` και το στοιχείο του πίνακα, αν βρέθηκε, διαφορετικά, το μήνυμα “NOT FOUND”.
- Θα **καλεί** την συνάρτηση `binarysearchPThesiNumREcPointers`(), η οποία βρίσκει με δυαδική αναζήτηση και επιστρέφει τη θέση του `num` στον ταξινομημένο πίνακα, αν υπάρχει, καλώντας τον εαυτό της αναδρομικά και με τη χρήση δεικτών.

- Θα **εμφανίζει** τη θέση που βρέθηκε το `num` και το στοιχείο του πίνακα, αν βρέθηκε, διαφορετικά, το μήνυμα "NOT FOUND".
- Θα **δημιουργεί** μια τυχαία ακέραια τιμή `index` στο `[0, n-2]`.
- Θα **καλεί** την συνάρτηση `swapPin`() με την οποία θα ανταλλάσσει το περιεχόμενο 2 διαδοχικών θέσεων (`index`, `index+1`) του πίνακα.
- Θα **εμφανίζει** τα περιεχόμενα του πίνακα καλώντας τη συνάρτηση `showPin()`.

Το αρχείο με την συνάρτηση `main()` θα περιέχει και τις συναρτήσεις:

- `fillPin()`, η οποία γεμίζει έναν πίνακα ακεραίων με τυχαίες τιμές από το `1` έως το `10`.
- `showPin()`, η οποία εμφανίζει τα περιεχόμενα του πίνακα.
- `findThesiMax()`, η οποία θα επιστρέφει τη θέση του στοιχείου με τη μεγαλύτερη τιμή στον πίνακα.
- `findThesiMin()`, η οποία θα επιστρέφει τη θέση του στοιχείου με τη μικρότερη τιμή στον πίνακα.
- `countAvg()`, η οποία θα επιστρέφει πόσα στοιχεία του πίνακα είναι πάνω από τον μέσο όρο.
- `swapPin`(), η οποία ανταλλάσσει το περιεχόμενο 2 διαδοχικών θέσεων του πίνακα.
- `searchPThesiNum`(), η οποία βρίσκει και επιστρέφει τη θέση του `num` στον πίνακα, αν υπάρχει.
- `bubbleDesc`(), η οποία ταξινομεί τα στοιχεία του πίνακα με την μέθοδο bubble sort, αλλά σε φθίνουσα τάξη.
- `binarysearchPThesiNum`(), η οποία βρίσκει με δυαδική αναζήτηση και επιστρέφει τη θέση του `num` στον ταξινομημένο πίνακα, αν υπάρχει, με τη χρήση δεικτών.
- `binarysearchPThesiNumRec`(), η οποία βρίσκει με δυαδική αναζήτηση και επιστρέφει τη θέση του `num` στον ταξινομημένο πίνακα, αν υπάρχει, καλώντας τον εαυτό της αναδρομικά χωρίς τη χρήση δεικτών.
- `binarysearchPThesiNumREcPointers`(), η οποία βρίσκει με δυαδική αναζήτηση και επιστρέφει τη θέση του `num` στον ταξινομημένο πίνακα, αν υπάρχει, καλώντας τον εαυτό της αναδρομικά και με τη χρήση δεικτών.

Ενδεικτική Έξοδος Προγράμματος

```
Give an integer n > 5 : 10
p = 2 8 5 1 10 5 9 9 3 5
max = 10 thesiMax = 4
min = 1 thesiMin = 3
avg = 5.700000
count avg = 4
Give an integer num : 9
Found num = 9 in position 6, p[6] = 9
p = 10 9 9 8 5 5 5 3 2 1
Give an integer num : 5
Found num = 5 in position 4, p[4] = 5
Found num = 5 in position 4, p[4] = 5
Found num = 5 in position 4, p[4] = 5
swap p[0], p[1]
p = 9 10 9 8 5 5 5 3 2 1
Press any key to continue . . .
```

Άσκηση 11.2

Να γραφεί Πρόγραμμα C το οποίο **χωρίς τη χρήση αγκυλών**, ούτε στη `main()` ούτε στις συναρτήσεις, αλλά με τη χρήση **δεικτών – pointers** και τη χρήση της **μνήμης σωρού** με την εντολή `malloc()` θα κάνει τα παρακάτω:

- Θα **διαβάζει** μια τιμή για το `n > 5`.
- Θα **γεμίζει** έναν πίνακα ακεραίων `nxn` θέσεων (`my2DArray`) με τυχαίες τιμές στο `[1,10]` καλώντας τη συνάρτηση `fillPin2D()`.
- Θα **εμφανίζει** τα περιεχόμενα του πίνακα καλώντας τη συνάρτηση `showPin2D()`.
- **Καλώντας** τη συνάρτηση `findMeanLine()` θα **υπολογίζει** το μέσο Όρο των στοιχείων κάθε γραμμής του πίνακα `my2DArray`. Η συνάρτηση θα **επιστρέφει** έναν πίνακα μίας διάστασης με όνομα `avgLine[]`, κάθε στοιχείο του οποίου θα περιέχει το μέσο όρο κάθε γραμμής του πίνακα `my2DArray`.
- Να **εμφανίζει** τα στοιχεία του πίνακα `avgLine` καλώντας τη συνάρτηση `showPinDouble1D()`.
- **Καλώντας** τη συνάρτηση `findMeanCol()` θα **υπολογίζει** το μέσο Όρο των στοιχείων κάθε στήλης του πίνακα `my2DArray`. Η συνάρτηση θα **επιστρέφει** έναν πίνακα μίας διάστασης με όνομα `avgCol[]`, κάθε στοιχείο του οποίου θα περιέχει το μέσο όρο κάθε στήλης του πίνακα `my2DArray`.
- Να **εμφανίζει** τα στοιχεία του πίνακα `AVGCol` καλώντας τη συνάρτηση `showPinDouble1D()`.
- **Καλώντας** τη συνάρτηση `findDSum()` να **βρίσκει** το άθροισμα των δύο διαγωνίων του πίνακα `my2DArray` και να το **εμφανίζει**. Η εμφάνιση να γίνεται στην `main()`.

Το αρχείο με την συνάρτηση `main()` θα περιέχει και τις συναρτήσεις:

- `fillPin2D()`, η οποία γεμίζει έναν πίνακα ακεραίων `nxn` θέσεων με τυχαίες τιμές από το `1` έως το `10`.
- `showPin2D()`, η οποία εμφανίζει τα περιεχόμενα του πίνακα `my2DArray`.
- `showPinDouble1D()`, η οποία εμφανίζει τα περιεχόμενα πίνακα μιας διάστασης με στοιχεία πραγματικούς αριθμούς.
- `findMeanLine()`, η οποία επιστρέφει έναν πίνακα με τους Μέσους Όρους των στοιχείων κάθε γραμμής του πίνακα.
- `findMeanCol()`, η οποία επιστρέφει έναν πίνακα με τους Μέσους Όρους των στοιχείων κάθε στήλης του πίνακα.
- `findDSum()`, η οποία επιστρέφει το άθροισμα των στοιχείων των δύο διαγωνίων του πίνακα.

Ενδεικτική Έξοδος Προγράμματος

```
Give an integer n > 5 : 10
p =
 2  8  5  1 10  5  9  9  3  5
 6  6  2  8  2  2  6  3  8  7
 2  5  3  4  3  3  2  7  9  6
 8  7  2  9 10  3  8 10  6  5
 4  2  3  4  4  5  2  2  4  9
 8  5  3  8  8 10  4  2 10  9
 7  6  1  3  9  7  1  3  5  9
 7  6  1 10  1  1  7  2  4  9
10  4  5  5  7  1  7  7  2  9
 5 10  7  4  8  9  9  3 10  2
avgLine = 5.7 5.0 4.4 6.8 3.9 6.7 5.1 4.8 5.7 6.7
avgCol = 5.9 5.9 3.2 5.6 6.2 4.6 5.5 4.8 6.1 7.0
sum1D = 41
sum1D + sum2D = 95
Press any key to continue . . .
```

Άσκηση 11.3

Να γραφεί Πρόγραμμα, το οποίο **<u>χωρίς τη χρήση αγκυλών</u>**, ούτε στη `main()` ούτε στις συναρτήσεις, αλλά με τη χρήση **δεικτών – pointers** και τη χρήση της **μνήμης σωρού** με την εντολή `malloc()` θα κάνει τα παρακάτω:

Δηλώνει 2 μεταβλητές `yp1` και `yp2` τύπου `ypallhlos`, μια **δομή** (`struct`), η οποία περιλαμβάνει τα πεδία `Όνομα`, `Αριθμό Μητρώου`, `είδος πτυχίου`, `ώρες υπερωρίας` και `βασικό Μισθό`, τα οποία **γεμίζουν** για τους 2 υπαλλήλους με την κλήση της συνάρτησης `gemismaPedion()`, με την οποία **περνάει** σταθερές τιμές για τα πεδία του 1ου υπαλλήλου, τις οποίες και εμφανίζει με την **κλήση** της συνάρτησης `emfanishPedion()`, ενώ για τα πεδία του 2ου υπαλλήλου **διαβάζει** τιμές απ' το πληκτρολόγιο σε τοπικές μεταβλητές, τις οποίες μετά **περνάει** παραμετρικά με την κλήση της συνάρτησης `gemismaPedion()`, τις οποίες και **εμφανίζει** με την κλήση της συνάρτησης `emfanishPedion()` και **Υπολογίζει** και **Εμφανίζει** τον Τελικό Μισθό του κάθε υπαλλήλου με την **κλήση** των συναρτήσεων `findTMisthos()` και `returnTMisthos()`. Ο **Τελικός Μισθός** του κάθε Υπαλλήλου, θα προκύπτει από την πρόσθεση στο Βασικό Μισθό του ποσού των υπερωριών και του επιδόματος πτυχίου. Οι υπερωρίες πληρώνονται `20 ΕΥΡΩ` την ώρα, ενώ το επίδομα πτυχίου είναι `300 ΕΥΡΩ` για τον κάτοχο Διδακτορικού Τίτλου (`eidosPtyxiou = 1`), `150 ΕΥΡΩ` για τον κάτοχο Μεταπτυχιακού Τίτλου (`eidosPtyxiou = 2`), 100 ΕΥΡΩ για τον κάτοχο Πτυχίου ΑΕΙ (`eidosPtyxiou = 3`), `50 ΕΥΡΩ` για για τον κάτοχο Πτυχίου ΙΕΚ (`eidosPtyxiou = 4`) και `0 ΕΥΡΩ` για τον κάτοχο απολυτηρίου Λυκείου-ΕΠΑΛ (`eidosPtyxiou = 5`), Γυμνασίου (`eidosPtyxiou = 6`) και Δημοτικού (`eidosPtyxiou = 7`). Το επίδομα πτυχίου θα πρέπει να υπολογίζεται με τη χρήση της εντολής `switch`.
Στη συνέχεια **καλεί** τη συνάρτηση `findMaxTM()` με την οποία **βρίσκει** το **μεγαλύτερο** Τελικό Μισθό των υπαλλήλων `yp1` και `yp2` και **ανταλλάσσει** τα περιεχόμενα των μεταβλητών `yp1` και `yp2` στη `main()` και μετά **ανταλλάσσει** ξανά τα περιεχόμενα των μεταβλητών `yp1` και `yp2` με την κλήση της συνάρτησης `swapYp1Yp2()`.

- Ο Πίνακας χαρακτήρων για το όνομα θα πρέπει να είναι δείκτης.

Αλγόριθμος main()

1. Εισαγωγή Στοιχείων 1ου υπαλλήλου – κλήση συνάρτησης `gemismaPedion()`
2. Εμφάνιση στοιχείων 1ου υπαλλήλου – κλήση συνάρτησης `emfanihPedion()`
3. **Υπολογισμός - Εμφάνιση Τελικού Βαθμού 1ου** υπαλλήλου **με κλήση συνάρτησης `findTMisthos()`**
4. Εισαγωγή Στοιχείων 2ου υπαλλήλου – κλήση συνάρτησης `gemismaPedion()`
5. Εμφάνιση στοιχείων 2ου υπαλλήλου – κλήση συνάρτησης `emfanihPedion()`
6. **Υπολογισμός - Εμφάνιση Τελικού Βαθμού 2ου** υπαλλήλου **με κλήση συνάρτησης `returnTMisthos()`**
7. **Υπολογισμός - Εμφάνιση Μέγιστου Τελικού Βαθμού 1ου - 2ου** υπαλλήλου **με κλήση συνάρτησης `findMaxTM()`**
8. **Ανταλλαγή περιεχομένων 1ου - 2ου** υπαλλήλου
9. Εμφάνιση στοιχείων 1ου – 2ου υπαλλήλου – κλήση συνάρτησης `emfanihPedion()`
10. **Ανταλλαγή περιεχομένων 1ου - 2ου** υπαλλήλου – κλήση συνάρτησης `swapYp1Yp2()`
11. Εμφάνιση στοιχείων 1ου – 2ου υπαλλήλου – κλήση συνάρτησης `emfanihPedion()`

Το αρχείο με την συνάρτηση `main()` θα περιέχει και τις συναρτήσεις:

- `returnTMisthos()`, η οποία Υπολογίζει και Επιστρέφει τον Τελικό Μισθό του κάθε υπαλλήλου.
- `findTMisthos()`, η οποία Υπολογίζει και Εμφανίζει τον Τελικό Μισθό του κάθε υπαλλήλου.
- `gemismaPedion()`, η οποία περνάει παραμετρικά τιμές στα πεδία του κάθε υπαλλήλου.
- `emfanishPedion()`, η οποία Εμφανίζει τα πεδία ενός υπαλλήλου.
- `swapYp1Yp2()`, η οποία ανταλλάσσει τα περιεχόμενα 2 υπαλλήλων.
- `findMaxTM()`, η οποία βρίσκει το μεγαλύτερο Τελικό Μισθό των υπαλλήλων `yp1` και `yp2`.

Ενδεικτική Έξοδος Προγράμματος

```
ypallhlos yp1 :
name : ioannou
aM = 191234
ptyxio = 1
yperories = 7
basikosMisthos = 600.0
call findTMisthos() - tM = 1040.000000
Give name of ypallhlos 2 : Georgiou
Give aM of ypallhlos 2 : 12345
Give ptyxio of ypallhlos 2 : 2
Give yperories of ypallhlos 2 : 30
Give basikosMisthos of ypallhlos 2 : 1000
ypallhlos yp2 :
name : Georgiou
aM = 12345
ptyxio = 2
```

```
yperories = 30
basikosMisthos = 1000.0
call returnTMisthos() - tM = 1750.000000
TM yp2 = 1750.000000 > TM yp1 = 1040.000000
ypallhlos yp1 after swap yp1 yp2  :
name : Georgiou
aM = 12345
ptyxio = 2
yperories = 30
basikosMisthos = 1000.0

ypallhlos yp2 after swap yp1 yp2  :
name : ioannou
aM = 191234
ptyxio = 1
yperories = 7
basikosMisthos = 600.0
ypallhlos yp1 after call swapYp1Yp2() :
name : Georgiou
aM = 12345
ptyxio = 2
yperories = 30
basikosMisthos = 1000.0
ypallhlos yp2 after call swapYp1Yp2()  :
name : ioannou
aM = 191234
ptyxio = 1
yperories = 7
basikosMisthos = 600.0
Press any key to continue . . .
```

Άσκηση 11.4

Να γραφεί Πρόγραμμα, το οποίο **χωρίς τη χρήση αγκυλών**, ούτε στη `main()` ούτε στις συναρτήσεις, αλλά με τη χρήση **δεικτών – pointers** και τη χρήση της **μνήμης σωρού** με την εντολή `malloc()` θα κάνει τα παρακάτω:

- **Διαβάζει** την ακέραια τιμή μιας μεταβλητής `n` για το μέγεθος του πίνακα `yp[n]` `n` μεταβλητών τύπου `ypallhlos`. Εκτός από τα πεδία `Όνομα`, `Αριθμό Μητρώου`, `είδος πτυχίου`, `ώρες υπερωρίας` και `Βασικό Μισθό`, η δομή τύπου `ypallhlos` περιλαμβάνει και το πεδίο `tM` για τον `Τελικό Μισθό`. Τα πεδία (εκτός από τον `Τελικό Μισθό`) **γεμίζουν** για τον καθένα από τους `n` υπαλλήλους με την **κλήση** της συνάρτησης `gemismaPedion()`. Με την **κλήση** της συνάρτησης `setTMAll()`, η οποία **καλεί** τη συνάρτηση `returnTMisthos()`, **γεμίζουν** όλα τα πεδία των `n` υπαλλήλων του πίνακα, τα οποία και **εμφανίζει** με την κλήση της συνάρτησης `emfanishPedion()`. Μετά **δημιουργεί** έναν τυχαίο ακέραιο αριθμό `index` μεταξύ `0` και `n-2` και καλεί τη συνάρτηση `swapYpiYpi1()`, με την οποία **ανταλλάσσει** τα περιεχόμενα των `yp[index]`, `yp[index+1]` και **εμφανίζει** ξανά τα στοιχεία όλων των υπαλλήλων. Στη συνέχεια **καλεί** τη συνάρτηση `returnThesiMaxTM()` με την οποία **βρίσκει** τον υπάλληλο που έχει το μεγαλύτερο Τελικό Μισθό και **εμφανίζει** τα στοιχεία του.

- Ο Τελικός Μισθός του κάθε Υπαλλήλου θα προκύπτει από την πρόσθεση στο Βασικό Μισθό του ποσού των υπερωριών και του επιδόματος πτυχίου. Οι υπερωρίες πληρώνονται `20 ΕΥΡΩ` την ώρα, ενώ το επίδομα πτυχίου είναι `300 ΕΥΡΩ` για τον κάτοχο Διδακτορικού Τίτλου (`eidosPtyxiou = 1`), `150 ΕΥΡΩ` για τον κάτοχο Μεταπτυχιακού Τίτλου (`eidosPtyxiou = 2`), 100 ΕΥΡΩ για τον κάτοχο Πτυχίου ΑΕΙ (`eidosPtyxiou = 3`), `50 ΕΥΡΩ` για για τον κάτοχο Πτυχίου ΙΕΚ (`eidosPtyxiou = 4`) και `0 ΕΥΡΩ` για τον κάτοχο απολυτηρίου Λυκείου-ΕΠΑΛ (`eidosPtyxiou = 5`), Γυμνασίου (`eidosPtyxiou = 6`) και Δημοτικού (`eidosPtyxiou = 7`).
- Το επίδομα πτυχίου θα πρέπει να υπολογίζεται με τη χρήση της εντολής `switch`
- Ο Πίνακας χαρακτήρων για το όνομα θα πρέπει να είναι **δείκτης**.

Αλγόριθμος main()

1. Διάβασμα τιμής `<= 50` στη μεταβλητή `n`.
2. Εισαγωγή Στοιχείων `n` υπαλλήλων – κλήση συνάρτησης `gemismaPedion()`
3. **Γέμισμα πεδίου** `tM` **των** `n` υπαλλήλων - **κλήση συνάρτησης** `setTMAll()`
4. Εμφάνιση στοιχείων `n` υπαλλήλων – κλήση συνάρτησης `emfanishPedion()`
5. Δημιουργία τυχαίου ακεραίου `index` στο `[0, n-2]`.
6. **Ανταλλαγή περιεχομένων** `yp[index], yp[index+1]` – κλήση συνάρτησης `swapYpiYpi1()`
7. Εμφάνιση στοιχείων `n` υπαλλήλων – κλήση συνάρτησης `emfanihPedion()`
8. **Εύρεση** υπαλλήλου **με Μέγιστο Τελικό Μισθό - κλήση συνάρτησης** `returnThesiMaxTM()`
9. **Εμφάνιση** στοιχείων υπαλλήλου **με Μέγιστο Τελικό Μισθό.**

Το αρχείο με την συνάρτηση `main()` θα περιέχει και τις συναρτήσεις:

- `returnTMisthos()`, η οποία Υπολογίζει και Επιστρέφει τον Τελικό Μισθό του κάθε υπαλλήλου.
- `setTMAll()`, η οποία Υπολογίζει τον Τελικό Μισθό για όλους τους υπαλλήλους.
- `gemismaPedion()`, η οποία περνάει παραμετρικά τιμές στα πεδία του κάθε υπαλλήλου.
- `emfanishPedion()`, η οποία Εμφανίζει τα πεδία ενός υπαλλήλου.
- `swapYp1Yp2()`, η οποία ανταλλάσσει τα περιεχόμενα 2 υπαλλήλων.
- `returnThesiMaxTM()`, η οποία βρίσκει τη θέση του υπαλλήλου στον Πίνακα με τον μεγαλύτερο Τελικό Μισθό.

Ενδεικτική Έξοδος Προγράμματος

```
Give n <= 50 : 3
Give name of ypallhlos yp[0] : ioannou
Give aM of ypallhlos yp[0] : 191234
Give ptyxio of ypallhlos yp[0] : 1
Give yperories of ypallhlos yp[0] : 12
Give basikosMisthos of ypallhlos yp[0] : 1000
Give name of ypallhlos yp[1]  : Georgiou
Give aM of ypallhlos yp[1]  : 12345
Give ptyxio of ypallhlos yp[1] : 2
Give yperories of ypallhlos yp[1] : 15
Give basikosMisthos of ypallhlos yp[1] : 600
Give name of ypallhlos yp[1]  : Antoniou
Give aM of ypallhlos yp[1]  : 23456
Give ptyxio of ypallhlos yp[1] : 3
Give yperories of ypallhlos yp[1] : 20
Give basikosMisthos of ypallhlos yp[1] : 500
ypallhlos yp[0] :
name : ioannou
aM = 191234
ptyxio = 1
yperories = 12
basikosMisthos = 1000.000000
telikosMisthos = 1540.000000
ypallhlos yp[1] :
name : Georgiou
aM = 12345
ptyxio = 2
yperories = 15
basikosMisthos = 600.000000
telikosMisthos = 1050.000000
ypallhlos yp[2] :
name : Antoniou
aM = 23456
ptyxio = 3
yperories = 20
basikosMisthos = 500.000000
telikosMisthos = 1000.000000
pinakas after Swap yp[0] <--> yp[1]
ypallhlos yp[0] :
name : Georgiou
aM = 12345
ptyxio = 2
yperories = 15
basikosMisthos = 600.000000
telikosMisthos = 1050.000000
ypallhlos yp[1] :
name : ioannou
aM = 191234
ptyxio = 1
yperories = 12
basikosMisthos = 1000.000000
telikosMisthos = 1540.000000
ypallhlos yp[2] :
name : Antoniou
aM = 23456
ptyxio = 3
yperories = 20
basikosMisthos = 500.000000
```

```
telikosMisthos = 1000.000000
Pedia ypallhlou yp[1] me maxTM :
name : ioannou
aM = 191234
ptyxio = 1
yperories = 12
basikosMisthos = 1000.000000
telikosMisthos = 1540.000000
Press any key to continue . . .
```

Οδηγίες κατάθεσης ασκήσεων

1. Συνδεθείτε στο URL: http://aetos.it.teithe.gr/s.
2. Επιλέξτε το μάθημα “Δομημένος Προγραμματισμός – Εργαστήριο Χ” (Όπου Χ ο αριθμός του εργαστηρίου του οποίου τις ασκήσεις πρόκειται να καταθέσετε) και πατήστε επόμενο
3. Συμπληρώστε τα στοιχεία σας. Πληκτρολογείστε USERNAME και PASSWORD ανάλογα με το τμήμα που παρακολουθείτε βάσει του παρακάτω πίνακα :

	T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	**T10**	**T11**	**T12**	**…**	**T29**
USERNAME	00001	00002	00003	00004	00005	00006	00007	00008	00009	00010	00011	00012	…	00029
PASSWORD	10000	20000	30000	40000	50000	60000	70000	80000	90000	10000	11000	12000	..	29000

4. Επιλέξτε το αρχείο που θέλετε να στείλετε επιλέγοντας “choose file” στο πεδίο FILE1 και πατήστε “Παράδοση”.