

Τμήμα Εφαρμοσμένης Πληροφορικής
ΔΙΑΔΙΚΑΣΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Εξάμηνο Α'

Φύλλο Ασκήσεων 8 – ΔΟΜΕΣ

Παρατηρήσεις:

1. Τα δεδομένα εισόδου διαβάζονται με τη σειρά που δηλώνονται στις εκφωνήσεις. Για κάθε δεδομένο εισόδου να χρησιμοποιείτε προτρεπτικό μήνυμα που θα ενημερώνει τον χρήστη για την τιμή που αναμένεται.
2. Αντίστοιχα για τα δεδομένα εξόδου και όπου δεν υπάρχουν περαιτέρω διευκρινήσεις για τη μορφή τους, αυτά θα εμφανίζονται με ξεχωριστές εντολές `printf("...\n")` το καθένα και με τη σειρά που δηλώνονται στις εκφωνήσεις.
3. Τα αριθμητικά δεδομένα αναπαρίστανται πάντα από μεταβλητές ακέραιου τύπου (`int` ή `long`). Σε αντίθετη περίπτωση (μεταβλητές τύπου `double`) θα γίνονται οι απαραίτητες διευκρινήσεις.
4. Η εμφάνιση τιμών τύπου `float` και `double` θα γίνεται με την εξής μορφοποίηση: `.1f` και `.1lf` αντίστοιχα, εκτός κι αν ορίζεται διαφορετικά στην άσκηση.
5. Χρήση του όρου «Επιστρέφει» μέσα σε εισαγωγικά: Στις ακόλουθες ασκήσεις όταν υπάρχει όρος «επιστρέφει», δεν σημαίνει κατά ανάγκη ότι οι τιμές που υπολογίζονται από την κληθείσα συνάρτηση θα επιστρέφονται με την εντολή `return`. Απαιτείται η κληθείσα συνάρτηση να υπολογίζει τις τιμές και να ενημερώνει κατάλληλα την καλούσα συνάρτηση (πχ με χρήση δεικτών).

1. Να γραφούν οι δηλώσεις τύπων για κάθε μια από τις παρακάτω δομές:

- (a) χρόνος αν τον μετράμε σε ώρες, λεπτά, δευτερόλεπτα
- (b) μήκος αν το μετράμε σε m, cm, mm
- (c) τηλεφωνικός κατάλογος (επώνυμο, όνομα, διεύθυνση, τηλέφωνο)
- (d) περιγραφή ενός αυτοκινήτου (κατασκευαστής, μοντέλο, χρώμα, αριθμός κυκλοφορίας, έτος κυκλοφορίας)
- (e) περιγραφή ενός βιβλίου στην βιβλιοθήκη (τίτλος, συγγραφέας, εκδότης, ISBN)
- (f) στοιχεία φοιτητή (επώνυμο, όνομα, ΑΜ, οδός-αριθμός, πόλη, ΤΚ, τηλέφωνο)
- (g) ημερομηνία (όνομα μήνα, έτος, αριθμός ημέρας)
- (h) πληροφορίες για το στοκ ενός είδους (όνομα είδους 20 χαρακτήρες, εξανήφιος αριθμός που δηλώνει το πλήθος των αντικειμένων που είναι διαθέσιμα, χρώμα, ημερομηνία στην μορφή (g))

2. Για τους ενοίκους μιας οικοδομής δίνονται τα παρακάτω στοιχεία:

- αριθμός διαμερίσματος (`int`)
- ονοματεπώνυμο (`char[25]`)
- τετραγωνικά μέτρα διαμερίσματος (`int`)
- φέτες καλοριφέρ (`int`)

Να γραφεί πρόγραμμα που θα εκτελεί τις παρακάτω συναρτήσεις:

1. τη συνάρτηση `ReadNumApparts` για το διάβασμα του πλήθους (`int`) `m` των διαμερισμάτων της οικοδομής (θα επιστρέφει το πλήθος των διαμερισμάτων).
2. Τη συνάρτηση `ReadMonthData` για το διάβασμα των μηνιαίων δαπανών για το ποσό θέρμανσης - `float`, και το ποσό κοινοχρήστων - `float`). Ο επιστρεφόμενος τύπος της συνάρτησης θα είναι τύπου δομής (`struct`)
3. Τη συνάρτηση `ReadEnoikoiData` για το διάβασμα των στοιχείων κάθε διαμερίσματος. Ο αριθμός διαμερίσματος θα δίνεται από το πρόγραμμα κατά αύξουσα σειρά. Τα παραπάνω δεδομένα για τα διαμερίσματα της οικοδομής θα εισάγονται σε πίνακα δομών (μέγιστη διάσταση πίνακα `N=20`). Στην `i` θέση του πίνακα θα καταχωρούνται τα στοιχεία του `i` διαμερίσματος. Ο αριθμός διαμερίσματος θα δίνεται αυτόματα από το πρόγραμμα και όχι από τον χρήστη (για το `i` διαμέρισμα θα δίνεται ως αριθμός διαμερίσματος το `i`).
4. Τη συνάρτηση `Calculations` για τον υπολογισμό των ποσών αναλυτικής κατάστασης πληρωμής. Θα υπολογίζει τα ποσά που θα πληρώσει κάθε διαμέρισμα για θέρμανση και κοινοχρήστα αλλά και τα συνολικά τετραγωνικά μέτρα της οικοδομής, το σύνολο των φετών της οικοδομής.
Το ποσό της θέρμανσης για κάθε διαμέρισμα υπολογίζεται ως εξής

(φέτες καλοριφέρ διαμερίσματος/σύνολο φετών όλων των διαμερισμάτων)*μηνιαίο ποσό για δαπάνες θέρμανσης

το ποσό κοινοχρήστων για κάθε διαμέρισμα υπολογίζεται ως εξής:

(τετραγωνικά μέτρα διαμερίσματος/σύνολο τετραγωνικών μέτρων όλων των διαμερισμάτων)*μηνιαίο ποσό για δαπάνες κοινοχρήστων

Επίσης θα υπολογίζει τα συνολικά ποσά που θα εισπραχθούν: για τη θέρμανση της οικοδομής, το συνολικό ποσό για τα κοινόχρηστα, και το σύνολο (είναι τα ποσά που φαίνονται στην τελευταία γραμμή του στιγμιότυπου που ακολουθεί). Ο επιστρεφόμενος τύπος της συνάρτησης θα είναι τύπου δομής (struct) που θα αποθηκεύει τα παραπάνω 3 συνολικά ποσά.

5. Συνάρτηση για την εμφάνιση της αναλυτικής κατάστασης πληρωμής.

Η αναλυτική κατάσταση θα εμφανίζεται σε δεξιά στοίχιση σύμφωνα με το παρακάτω πρότυπο:

1-3	4-29	30-34	35-40	41-52	53-64	65-76
A/A	ONOMATEPWNIMO	T.M.	FETES	POSO THERM.	POPO KOIN.	SYNOLO
1	APETH MAPΔA	125	50	10000	12000	22000
.....
SYNOLA:	

Η main() θα καλεί τις παραπάνω συναρτήσεις. Δίνεται ένα στιγμιότυπο εκτέλεσης.

```
DWSE TO PLH8OS TWN DIAMERISMATWN?2
DWSE TO MHNIAIO POSO THERMANSHS? 1000
DWSE TO MHNIAIO POSO KOIMOXRHSTWN? 600
Dwse onomatepwnymo? ASPASIDOU
Dwse t.m? 100
Dwse fetes? 40
Dwse onomatepwnymo? GEWRGIU
Dwse t.m? 50
Dwse fetes? 20
A/A          ONOMATEPWNIMO T.M. FETES POSO THERM.  POSO KOIN.      SYNOLO
-----
1            ASPASIDOU    100   40    666.67    400.00    1066.67
2            GEWRGIU      50    20    333.33    200.00    533.33
-----
SYNOLA:          150    60    1000.00    600.00    1600.00
```

3. Για τα τιμολόγια που εκδόθηκαν από μια εταιρεία δίνονται:

- ημερομηνία έκδοσης τιμολογίου (HH/MM/EEEE) char[11]
- αριθμός τιμολογίου (int)
- καθαρή αξία εμπορευμάτων (long)
- συντελεστής ΦΠΑ (6%, 13%, 24%), μεταβλητή τύπου int (π.χ. 4)

Να γραφεί πρόγραμμα που θα εκτελεί τις παρακάτω λειτουργίες:

1. Εισαγωγή του τρέχοντος έτους (μεταβλητή τύπου int) στη main().

2. Εισαγωγή του πλήθους (num) τιμολογίων στη main().

3. Εισαγωγή στοιχείων τιμολογίων σε μια συνάρτηση με το εξής πρωτότυπο:

```
int read_timologia(int num, int year, timrec timologia[]);
```

που δέχεται το πλήθος num των τιμολογίων, το τρέχον έτος year και τον πίνακα εγγραφών timologia (timrec είναι ο τύπος της εγγραφής τιμολογίου) τον οποίο ενημερώνει. Η συνάρτηση επιστρέφει το πλήθος των έγκυρων τιμολογίων.

4. Εμφάνιση στοιχείων τιμολογίων σε μια διαδικασία με το εξής πρωτότυπο:

```
void write_data(int count, timrec timologia[]);
```

όπου count είναι ο αριθμός των έγκυρων τιμολογίων που καταχωρήθηκαν στον πίνακα εγγραφών timologia.

5. Υπολογισμός ποσών για κάθε κατηγορία ΦΠΑ ανά δίμηνο σε μια συνάρτηση με το εξής πρωτότυπο:

```
resrec calculate_results(int m, timrec timologia[], resrec result[]);
```

που δέχεται το πλήθος m των τιμολογίων, τον πίνακα εγγραφών timologia (timrec είναι ο τύπος της εγγραφής τιμολογίου) και τον πίνακα εγγραφών result που ενημερώνεται με τα ποσά για κάθε κατηγορία ΦΠΑ ανά δίμηνο. Η συνάρτηση επιστρέφει μια εγγραφή τύπου resrec με το γενικό σύνολο για κάθε κατηγορία ΦΠΑ ετησίως.

6. Εμφάνιση συνολικών ποσών καθαρής αξίας, ΦΠΑ για κάθε κατηγορία και γενικού συνόλου ανά δίμηνο κι ετησίως, χρησιμοποιώντας μια συνάρτηση με το εξής πρωτότυπο:

```
void write_results(resrec results[], resrec sum);
```

Τα στοιχεία των τιμολογίων θα καταχωρούνται σε πίνακα εγγραφών (διάσταση πίνακα N=20, και num≤N). Κατά την εισαγωγή των στοιχείων θα ελέγχεται η εγκυρότητα της ημερομηνίας έκδοσης. **Σε περίπτωση μη εγκυρότητας της ημερομηνίας δε θα καταχωρείται η εγγραφή τιμολογίου στον πίνακα.** Σε κάθε περίπτωση ο χρήστης θα εισάγει στοιχεία για num τιμολόγια είτε τα στοιχεία είναι έγκυρα οπότε θα καταχωρούνται και στον πίνακα είτε δεν είναι έγκυρα οπότε δεν καταχωρούνται στον πίνακα. Η εμφάνιση των στοιχείων των τιμολογίων θα γίνεται σε δεξιά στοίχιση ως εξής:

1-10	11-15	16-25	26-34
ΗΜΕΡΟΜΗΝΙΑ	A.T.	KATH.AXIA	SYN.FPA
12/11/1999	153	100000	0.04
.....

Η εμφάνιση των συνολικών ποσών ανά δίμηνο κι ετησίως και των τελικών συνόλων θα γίνεται σε δεξιά στοίχιση ανά κατηγορία ΦΠΑ ως εξής:

1-7	8-18	19-27	28-36	37-45	46-56
DI.	SYN.K.AXIA	FPA 4%	FPA 8%	FPA 18%	SYNOLO
1	112000	4400	80	180	116660
..
SYNOLA:

Η άσκηση να λυθεί:

- χρησιμοποιώντας συναρτήσεις, οι οποίες θα καλούνται από το κυρίως πρόγραμμα σειριακά
- η καταχώρηση των στοιχείων των τιμολογίων στον πίνακα θα συνεχίζεται μέχρι ο χρήστης να δώσει στοιχεία για num τιμολόγια. Τα τιμολόγια με μη έγκυρη ημερομηνία δεν θα καταχωρούνται στον πίνακα και κατά συνέπεια δεν θα εμφανίζονται ούτε θα συμμετέχουν στους υπολογισμούς, δηλαδή είναι ενδεχόμενο αν δοθούν μη έγκυρα στοιχεία το πλήθος των τιμολογίων που θα καταχωρηθούν τελικώς στον πίνακα να είναι μικρότερο του num.
- μεταξύ των εμφανίσεων των διαδικασιών 4 και 6 δε θα υπάρχει κενή γραμμή

4. Σε ένα γραφείο ταχυδρομικών αποστολών καταχωρούνται καθημερινά διάφορες κινήσεις. Για κάθε αποστολή δίνονται οι παρακάτω πληροφορίες:

- Ονοματεπώνυμο (τύπου char[26])
- Διεύθυνση παραλήπτη (οδός και αριθμός, τύπου char[26])
- Ταχ. Κώδικας (τύπου int)
- Πόλη (τύπου char[16])
- Χώρα (τύπου char[16])
- Τύπος δέματος (1 εσωτερικού, 2 εξωτερικού) (int)
- Βάρος (τύπου int)

Να γραφεί πρόγραμμα το οποίο θα εκτελεί τις παρακάτω συναρτήσεις:

- συνάρτηση για το διάβασμα των στοιχείων των M ταχυδρομικών αποστολών.
- συνάρτηση για τον υπολογισμό του πλήθους των δεμάτων (int) και του συνολικού κόστους αποστολής για δέματα εσωτερικού κι εξωτερικού (float). Για τον υπολογισμό του κόστους αποστολής να λάβετε υπόψη ότι το κιλό βάρους στοιχίζει 0.40€ για δέματα εσωτερικού και 0.84€ για δέματα εξωτερικού. Το

κόστος αποστολής υπολογίζεται ως εξής:

για δέμα εσωτερικού: βάρος $X 0.40$

για δέμα εξωτερικού: βάρος $X 0.84$

3. Συνάρτηση για την εμφάνιση του πλήθους των δεμάτων και των συνολικών ποσών για αποστολές εσωτερικού κι εξωτερικού.

Το πλήθος των ταχυδρομικών αποστολών M θα διαβάζεται στη `main`. Τα στοιχεία των M ταχυδρομικών αποστολών θα καταχωρούνται σε μονοδιάστατο πίνακα N εγγραφών (διάσταση πίνακα $N=20$ και $M \leq N$, δε χρειάζεται να γίνεται έλεγχος εγκυρότητας της τιμής του M).

Η εμφάνιση των αποτελεσμάτων θα γίνει όπως φαίνεται στο στιγμιότυπο εκτέλεσης

```
Dwse to plhthos twn dematwn? 3
```

```
Name: ASPA
```

```
Address: EGNATIA 156
```

```
Zip: 54006
```

```
City: THESSALONIKI
```

```
Country: GREECE
```

```
Type (1 or 2): 1
```

```
Weight: 1
```

```
Name: NIKOS
```

```
Address: TSIMISKI 20
```

```
Zip: 54000
```

```
City: THESSALONIKI
```

```
Country: GREECE
```

```
Type (1 or 2): 1
```

```
Weight: 2
```

```
Name: SASA
```

```
Address: OXFORD 2
```

```
Zip: 44444
```

```
City: LONDON
```

```
Country: UK
```

```
Type (1 or 2): 2
```

```
Weight: 3
```

```
ESWTERIKOU: 2 1.20
```

```
EXWTERIKOU: 1 2.52
```

5. Πίνακας εγγραφών (με μέγιστο μέγεθος $N=20$) περιέχει τα παρακάτω στοιχεία για πωλητές:

- Κωδικός (`int`)
- Επώνυμο (`char[16]`)
- ποσό πωλήσεων (`long`)
- περιοχή (`int`) (έδρα εργασίας του υπαλλήλου, 1= Thes/niki, 2= Athens, 3= Volos, 4= Hrakleio) (`int`)

Να γραφεί πρόγραμμα που θα εκτελεί τις εξής λειτουργίες:

1. Εισαγωγή του πλήθους (`int`) `count` των πωλητών στη `main()`.
2. Εισαγωγή των στοιχείων των `count` πωλητών στον πίνακα εγγραφών, χρησιμοποιώντας μια συνάρτηση.
3. Υπολογισμός των συνολικών ποσών πωλήσεων ανά περιοχή (`long`) σε μια συνάρτηση. Τα συνολικά ποσά να καταχωρούνται σε ένα μονοδιάστατο πίνακα.
4. Εμφάνιση των συνολικών ποσών πωλήσεων ανά περιοχή σε μια συνάρτηση. Η εμφάνιση θα έχει ως εξής:

1-11	12-24
PERIOXH SYN.PWLHSEWN	

Thes/niki	18500
Athens
Volos
Hrakleio

SYNOLO:	18500

Οι συναρτήσεις θα καλούνται από το κυρίως πρόγραμμα σειριακά.

```

Dwse to plthos tw pwlhtwn? 5
Kwdikos? 123
Onomatepwynmo? Nikolaou
Poso pwlhsewn? 10000
Kwdikos (1= Thes/niki, 2= Athens, 3= Volos, 4= Hrakleio)? 1
Kwdikos? 431
Onomatepwynmo? Papadopoulou
Poso pwlhsewn? 4000
Kwdikos (1= Thes/niki, 2= Athens, 3= Volos, 4= Hrakleio)? 2
Kwdikos? 108
Onomatepwynmo? Petridis
Poso pwlhsewn? 6000
Kwdikos (1= Thes/niki, 2= Athens, 3= Volos, 4= Hrakleio)? 3
Kwdikos? 345
Onomatepwynmo? Petrou
Poso pwlhsewn? 3500
Kwdikos (1= Thes/niki, 2= Athens, 3= Volos, 4= Hrakleio)? 1
Kwdikos? 768
Onomatepwynmo? Nikou
Poso pwlhsewn? 3000
Kwdikos (1= Thes/niki, 2= Athens, 3= Volos, 4= Hrakleio)? 2
PERIOXH SYN.PWLHSEWN
-----
Thes/niki      13500
Athens         7000
Volos          6000
Hrakleio       0
-----
SYNOLO:        26500

```

6. Για κάθε ασθενή μιας κλινικής διατηρούνται τα εξής στοιχεία: επώνυμο (char [20]), όνομα (char [15]), διεύθυνση (char [30]) και ποσό πληρωμής (int). Τα παραπάνω δεδομένα εισάγονται σε πίνακα δομών (μέγιστη διάσταση πίνακα N=50), ο οποίος θα αφορά την κλινική.

Να γραφεί πρόγραμμα που θα εκτελεί τις παρακάτω συναρτήσεις:

1. Εισαγωγή των στοιχείων ασθενών (παράμετροι συνάρτησης: πίνακας ασθενών, πλήθος τους)
2. Εμφάνιση όλων των ασθενών της κλινικής (αναλυτική κατάσταση) (παράμετροι συνάρτησης: όνομα κλινικής, πίνακας ασθενών, πλήθος τους)
3. Εύρεση και επιστροφή του ασθενή με το μεγαλύτερο ποσό (παράμετροι συνάρτησης: πίνακας ασθενών, πλήθος τους).
4. Υπολογισμός των συνολικών εισπράξεων της κλινικής (παράμετροι συνάρτησης: πίνακας ασθενών, πλήθος τους. Επιστρέφει το συνολικό ποσό εισπράξεων)
5. Εμφάνιση ενός ασθενή (όλα τα στοιχεία του ασθενή) (παράμετροι συνάρτησης: ο ασθενής)
6. Αναζήτηση ασθενή (παράμετροι συνάρτησης: πίνακας ασθενών, πλήθος τους, ο ασθενής. Επιστρέφει TRUE (Boolean) αν βρέθηκε ο ασθενής και FALSE σε αντίθετη περίπτωση). Η αναζήτηση θα γίνεται με το επώνυμο και το όνομα που θα διαβάζονται (δίνονται από το χρήστη) εντός της συνάρτησης. Τα πλήρη στοιχεία του ασθενή θα "επιστρέφονται" από την αντίστοιχη παράμετρο της συνάρτησης.

Η εμφάνιση (συναρτήσεις 2 & 5) θα γίνονται με αριστερή στοίχιση σύμφωνα με το παρακάτω παράδειγμα εκτέλεσης.

Ελέγξτε τις παραπάνω συναρτήσεις για 2 κλινικές πχ την κλινική GALINOS με 3 ασθενείς και την ELPIS με 2 ασθενείς (για κάθε κλινική θα δημιουργήσετε ξεχωριστό πίνακα). Το πρόγραμμα θα εμφανίζει τον ασθενή με το μεγαλύτερο ποσό πληρωμής για κάθε κλινική, και την κλινική με τα μεγαλύτερα έσοδα (όνομα κλινικής και το ποσό). Επίσης θα αναζητά 1 ασθενή στην κλινική GALINOS και 1 ασθενή στην κλινική ELPIS. Δίνεται το παρακάτω δείγμα εκτέλεσης:

```

Give data patient 0
Last name: ASPASIDOU
First name: ASPA
Address: NEMEAS 8
Amount: 1000

Give data patient 1
Last name: GEORGIOU
First name: GEORGIOS
Address: PILEAS 10
Amount: 200

Give data patient 2
Last name: ADREOU

```

```

First name: ADREAS
Address: OLGAS 17
Amount: 230

Give data patient 0
Last name: PANTOS
First name: NIKOS
Address: ALMUROU 21
Amount: 89

Give data patient 1
Last name: ALEXIOU
First name: ALEXIOS
Address: TSIMISKH 33
Amount: 125
-----GALINOS-----
Last name      First Name      Address      Amount
-----
ASPASIDOU      ASPA      NEMEAS 8      1000
GEORGIOU      GEORGIOS      PILEAS 10      200
ADREOU      ADREAS      OLGAS 17      230
-----ELPIS-----
Last name      First Name      Address      Amount
-----
PANTOS      NIKOS      ALMUROU 21      89
ALEXIOU      ALEXIOS      TSIMISKH 33      125
Last name      First Name      Address      Amount
-----
ASPASIDOU      ASPA      NEMEAS 8      1000
Last name      First Name      Address      Amount
-----
ALEXIOU      ALEXIOS      TSIMISKH 33      125
Max income GALINOS 1430
Search a patient for GALINOS
Give last name ADREOU
Give first name ADREAS
Last name      First Name      Address      Amount
-----
ADREOU      ADREAS      OLGAS 17      230
Search a patient for ELPIS
Give last name PANTOS
Give first name NIKOS
Last name      First Name      Address      Amount
-----
PANTOS      NIKOS      ALMUROU 21      89

```

7. Μία εταιρία ενοικίασης αυτοκινήτων διατηρεί για την κάθε ενοικίαση τα εξής στοιχεία:

- Αριθμός ενοικίασης
- Μάρκα
- Κυβικά
- Όνομα ενοικιαστή
- Ημέρες ενοικίασης
- Τιμή ανά ημέρα

Να γραφεί πρόγραμμα που θα εκτελεί τις παρακάτω λειτουργίες:

1. Εισαγωγή του πλήθους των ενοικιάσεων
2. Εισαγωγή των στοιχείων κάθε ενοικίασης
3. Υπολογισμός αναλυτικής κατάστασης πληρωμής.
4. Υπολογισμός της ενοικίασης με το μεγαλύτερο κέρδος
5. Εμφάνιση αναλυτικής κατάστασης
6. Εμφάνιση των στοιχείων (Μάρκα, Κυβικά, Συνολικό Ποσό) για την ενοικίαση με το μεγαλύτερο κέρδος.

Τα παραπάνω δεδομένα θα εισάγονται σε πίνακα δομών (μέγιστη διάσταση πίνακα N=20). Στην *i* θέση του πίνακα θα καταχωρούνται τα στοιχεία του *i* αυτοκινήτου. Ο αριθμός ενοικίασης θα δίνεται αυτόματα από το πρόγραμμα και όχι από τον χρήστη (για την *i* ενοικίαση θα δίνεται ως αριθμός ενοικίασης το *i*). Η αναλυτική κατάσταση θα εμφανίζεται σε αριστερή στοίχιση σύμφωνα με το παρακάτω παράδειγμα εκτέλεσης:

```
Dose ton arithmo ton enoikiaseon: 3

Dose ta stoixeia tis enoikiasis 0
Dose marka: Audi
Dose kyvika: 2500
Dose onoma pelati: Papadopoulos
Dose imeres enoikiasis: 1
Dose timi ana imera: 100

Dose ta stoixeia tis enoikiasis 1
Dose marka: Fiat
Dose kyvika: 1200
Dose onoma pelati: Nikolaou
Dose imeres enoikiasis: 20
Dose timi ana imera: 10

Dose ta stoixeia tis enoikiasis 2
Dose marka: Alfa Romeo
Dose kyvika: 1800
Dose onoma pelati: Dimitriou
Dose imeres enoikiasis: 5
Dose timi ana imera: 30
```

Number	Name	Type	CC	Days	Price	Total
0	Papadopoulos	Audi	2500	1	100.00	100.00
1	Nikolaou	Fiat	1200	20	10.00	200.00
2	Dimitriou	Alfa Romeo	1800	5	30.00	150.00
					Total	450.00

```
Best car: Fiat 1200 rented for 200.00 Euros.
```

8. Να γραφεί ένα πρόγραμμα που να περιλαμβάνει μια δομή **box**, η οποία αναπαριστά ορθογώνια παραλληλεπίπεδα, και περιλαμβάνει πέντε ακέραια πεδία: **width** (πλάτος), **height** (ύψος), **depth** (βάθος), **area** (επιφάνεια) και **volume** (όγκος). Το πρόγραμμα θα περιλαμβάνει τις παρακάτω συναρτήσεις:

- Μια συνάρτηση που να ζητά από το χρήστη τις διαστάσεις του κουτιού σε cm (μη-αρνητικοί ακέραιοι – δεν απαιτείται έλεγχος), δηλαδή το πλάτος, ύψος και βάθος (*width*, *height*, *depth*) και θα επιστρέφει δομή **box** με τα αντίστοιχα τρία πεδία συμπληρωμένα.
- Μια συνάρτηση **void calc_box_area(box *a_box)** που να υπολογίζει το συνολικό εμβαδόν όλων των πλευρών του κουτιού (πεδίο **area**). Η συνάρτηση θα ενημερώνει το αντίστοιχο πεδίο της δομής.
- Μια συνάρτηση **void calc_box_volume(box *a_box)** που να υπολογίζει τον όγκο (πεδίο **volume**) του κουτιού. Η συνάρτηση θα ενημερώνει το αντίστοιχο πεδίο της δομής.
- Οι δύο παραπάνω τιμές (*area* και *volume*) θα τυπώνονται από τη συνάρτηση **main**.

Και οι δύο συναρτήσεις να δέχονται ως παράμετρο έναν δείκτη στη δομή **box**, όπως φαίνεται παραπάνω. Παράδειγμα εκτέλεσης δίνεται παρακάτω.

Παράδειγμα εκτέλεσης 1

```
Dose to mikos tou koutiou se cm: 10
Dose to ypsos tou koutiou se cm: 10
Dose to vathos tou koutiou se cm: 10
To emvadon tou koutiou einai 600 cm2
O ogos tou koutiou einai 1000 cm3
```

Παράδειγμα Εκτέλεσης 2:

```
Dose to mikos tou koutiou se cm: 5
Dose to ypsos tou koutiou se cm: 10
Dose to vathos tou koutiou se cm: 15
To emvadon tou koutiou einai 550 cm2
O ogos tou koutiou einai 750 cm3
```

9. Ένα φροντιστήριο ξένων γλωσσών διατηρεί τα παρακάτω στοιχεία μισθοδοσίας των εργαζομένων (καθηγητών).

- Όνομα και Επώνυμο
- Μισθός ανά ώρα εργασίας
- Ώρες εργασίας
- Μικτές αποδοχές
- Κρατήσεις
- Φόρος
- Καθαρές αποδοχές

Να γραφεί πρόγραμμα το οποίο θα ορίζει μία δομή υπαλλήλου με τα παραπάνω στοιχεία και θα εκτελεί τις κάτωθι λειτουργίες:

1. εισαγωγή του πλήθους των υπαλλήλων
2. εισαγωγή στοιχείων καθηγητή
3. εισαγωγή μισθού
4. εισαγωγή ωρών εργασίας
5. Υπολογισμός μικτών αποδοχών
6. Υπολογισμός καθαρών αποδοχών
7. Εμφάνιση αναλυτικής κατάστασης

Να ομαδοποιηθούν κατάλληλα οι παραπάνω λειτουργίες (εκτός της 1) σε συναρτήσεις:

- Συνάρτηση **get_data**, λειτουργίες 2,3,4
- Συνάρτηση **calc_salaries** λειτουργίες 5,6
- Συνάρτηση **print_data** λειτουργία 7.

Οι κρατήσεις είναι το 15% των μικτών αποδοχών και ο φόρος είναι 7% των μικτών μείον τις κρατήσεις.

```
Dose ton arithmo ton kathigiton: 2
Dose ta stoiceia tou kathigiti 0
Dose onoma: Nikos
Dose eponymo: Ioannou
Dose mistho oras: 12
Dose ores ergasias: 20
Dose ta stoiceia tou kathigiti 1
Dose onoma: Petros
Dose eponymo: Papadopoulos
Dose mistho oras: 25
Dose ores ergasias: 25
```

Name	Surname	Hourly Rate	Hours Worked	Gross	Deductions	Tax	Net
Nikos	Ioannou	12.00	20	240.00	36.00	14.28	189.72
Petros	Papadopoulos	25.00	25	625.00	93.75	37.19	494.06

```
Press any key to continue . . .
```

10. Σε ένα ημερολόγιο ενός έργου καταγράφονται διάφορα πακέτα εργασίας. Κάθε εργασία έχει όνομα **name** (**char name [60]**), μήνα έναρξης (**int**) όπως μετράται από την αρχή του έργου και διάρκεια σε μήνες (**int**). Για παράδειγμα:

- η εργασία “project specifications” έχει χρόνο έναρξης 0 (ξεκινά τον πρώτο μήνα του έργου) και διάρκεια 4 μήνες, οπότε θεωρείται ότι ολοκληρώνεται τον 4 μήνα.
- η εργασία “implementation” έχει χρόνο έναρξης 2 και διάρκεια 8 μήνες, οπότε θεωρείται ότι ολοκληρώνεται τον 10 μήνα ($10 = 8 + 2$)

(α) Να ορίσετε κατάλληλη δομή (νέο τύπο) η οποία να αποθηκεύει τα στοιχεία των εργασιών.

(β) Να ορίσετε ένα πρόγραμμα το οποίο:

- Ζητά από τον χρήστη τον συνολικό αριθμό εργασιών του έργου **WP**
- Μέσω μιας συνάρτησης **ReadWorkpackages** ζητά από τον χρήστη τα στοιχεία των **WP** εργασιών (όνομα, χρόνο έναρξης και διάρκεια)
- Βρίσκει ποια εργασία ξεκινά πρώτη μέσω της συνάρτησης **FindFirstWp**, η οποία επιστρέφει την δομή της πρώτης χρονικά εργασίας.

- Τυπώνει τα στοιχεία της πρώτης χρονικά εργασίας στην οθόνη μέσω της συνάρτησης **main()**.
- Βρίσκει την συνολική διάρκεια του έργου (*makespan*) μέσω της συνάρτησης **FindMakespan**. Η συνολική διάρκεια του έργου ορίζεται ως ο μήνας στον οποίο ολοκληρώνεται η τελευταία εργασία (έχουν τελειώσει όλες οι εργασίες).
- Εμφανίζει την συνολική διάρκεια του έργου στην οθόνη.

Για να γίνουν τα παραπάνω θα πρέπει να ορίσετε κατάλληλο πίνακα δομών με μέγεθος τουλάχιστον 100. Θεωρείστε ότι ο χρήστης (α) δεν θα εισάγει περισσότερες από 100 εργασίες (δεν απαιτείται έλεγχος), (β) θα εισάγει πάντα ως χρόνο έναρξης ένα μη-αρνητικό ακέραιο (δεν απαιτείται έλεγχος) (γ) ότι θα εισάγει πάντα ως διάρκεια ένα θετικό ακέραιο (δεν απαιτείται έλεγχος) και (δ) τα ονόματα των εργασιών είναι αυστηρά μικρότερα των 60 χαρακτήρων (δεν απαιτείται έλεγχος).

Τα μηνύματα εμφανίζονται στα παραδείγματα που ακολουθούν.

Παράδειγμα Εκτέλεσης 1

```
Total work packages: 2
WP Name:project specifications
WP Start Month:0
WP Duration:4
WP Name:implementation
WP Start Month:2
WP Duration:8
FIRST WP: project specifications start: 0 duration: 4
Total duration = 10
```

Παράδειγμα Εκτέλεσης 2

```
Total work packages:3
WP Name:debug
WP Start Month:5
WP Duration:8
WP Name:code
WP Start Month:3
WP Duration:6
WP Name:specs
WP Start Month:1
WP Duration:6
FIRST WP: specs start: 1 duration: 6
Total duration = 13
```

Παράδειγμα Εκτέλεσης 3

```
Total work packages:3
WP Name:ship
WP Start Month:8
WP Duration:2
WP Name:construct
WP Start Month:1
WP Duration:16
WP Name:buy
WP Start Month:6
WP Duration:2
FIRST WP: construct start: 1 duration: 16
Total duration = 17
```