

Τμήμα Εφαρμοσμένης Πληροφορικής
ΔΙΑΔΙΚΑΣΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Εξάμηνο Α'

Φύλλο Ασκήσεων 8 – ΔΟΜΕΣ

**Διδάσκοντες: Μάγια Σατρατζέμη, Αλέξανδρος Χατζηγεωργίου, Στέλιος Ξυνόγαλος,
Θεόδωρος Κασκάλης, Ηλίας Σακελλαρίου, Αλέξανδρος Καρακασίδης**

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

- χρόνος αν τον μετράμε σε ώρες, λεπτά, δευτερόλεπτα
- μήκος αν το μετράμε σε m, cm, mm
- τηλεφωνικός κατάλογος (επώνυμο, όνομα, διεύθυνση, τηλέφωνο)
- περιγραφή ενός αυτοκινήτου (κατασκευαστής, μοντέλο, χρώμα, αριθμός κυκλοφορίας, έτος κυκλοφορίας)
- περιγραφή ενός βιβλίου στην βιβλιοθήκη (τίτλος, συγγραφέας, εκδότης, ISBN)
- στοιχεία φοιτητή (επώνυμο, όνομα, ΑΜ, οδός-αριθμός, πόλη, ΤΚ, τηλέφωνο)
- ημερομηνία (όνομα μήνα, έτος, αριθμός ημέρας)
- πληροφορίες για το στοκ ενός είδους (όνομα είδους 20 χαρακτήρες, εξαψήφιος 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 λειτουργίες:

- Εισαγωγή του πλήθους (`int`) `m` των ενοίκων.
- Εισαγωγή μηνιαίων στοιχείων (ποσό θέρμανσης - `double`, ποσό κοινοχρήστων - `double`).
- Εισαγωγή στοιχείων ενοίκων. Ο αριθμός διαμερίσματος θα δίνεται από το πρόγραμμα κατά αύξουσα σειρά.
- Υπολογισμός ποσών αναλυτικής κατάστασης πληρωμής.
- Εμφάνιση αναλυτικής κατάστασης πληρωμής.

Τα παραπάνω δεδομένα θα εισάγονται σε πίνακα δομών (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`). 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
.....
SYNOLO:	

Η άσκηση να λυθεί χρησιμοποιώντας συναρτήσεις, οι οποίες θα καλούνται από το κυρίως πρόγραμμα σειριακά.

Dwse to plithos twn enoikwn?2						
Dwse to poso thermanshs? 1000						
Dwse to poso koinoxrhstwn? 600						
Dwse to onomatepwynymo? Aspasiou						
Dwse ta t.m? 100						
Dwse tis fetes? 40						
Dwse to onomatepwynymo? Gewrgiou						
Dwse ta t.m? 50						
Dwse tis fetes? 20						
A/A	ONOMATEPWYNIMO	T.M.	FETES	POSO THERM.	POSO KOIN.	SYNOLO
1	Aspasiou	100	40	666.67	400.00	1066.67
2	Gewrgiou	50	20	333.33	200.00	533.33
SYNOLOA:		150	60	1000.00	600.00	1600.00
Πιέστε ένα πλήκτρο για συνέχεια. . .						

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 (4%, 8%, 18%), 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
HMEROMHΝIA	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 `long`)
- Πόλη (τύπου `char[16]`)
- Χώρα (τύπου `char[16]`)
- Τύπος δέματος (1 εσωτερικού, 2 εξωτερικού) (`int`)
- Βάρος (τύπου `long`)

Να γραφεί πρόγραμμα το οποίο θα εκτελεί τις παρακάτω λειτουργίες:

- Εισαγωγή του πλήθους (`int`) M των ταχυδρομικών αποστολών.
- Εισαγωγή των στοιχείων των M ταχυδρομικών αποστολών σε μια συνάρτηση.
- Υπολογισμός των αριθμών δεμάτων (`int`) και των συνολικών ποσών εσωτερικού κι εξωτερικού (`double`) σε μια συνάρτηση.
- Εμφάνιση των αριθμών δεμάτων και των συνολικών ποσών εσωτερικού κι εξωτερικού σε μια συνάρτηση.

Τα στοιχεία των M ταχυδρομικών αποστολών θα καταχωρούνται σε μονοδιάστατο πίνακα N εγγραφών (διάσταση πίνακα N=20 και M≤N). Για τον υπολογισμό των ποσών αποστολής να λάβετε υπόψη ότι το κιλό βάρους στοιχίζει 0.40€ για δέματα εσωτερικού και 0.84€ για δέματα εξωτερικού.

Η εμφάνιση των αποτελεσμάτων θα έχει ως εξής:

συνολικός αριθμός δεμάτων εσωτερικού_συνολικό ποσό ταχ. αποστολών εσωτερικού
συνολικός αριθμός δεμάτων εξωτερικού_συνολικό ποσό ταχ. αποστολών εξωτερικού

```
Dwse to plithos 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: Londin
Country: GB
Type <1 or 2>: 2
Weight: 3
2 1.20
1 2.52
Πιέστε ένα πλήκτρο για συνέχεια. . .
```

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.PWLHSEΩN
Thes/niki	18500
Athens
Volos
Hrakleio
SYNOLO:	18500

Οι συναρτήσεις θα καλούνται από το κυρίως πρόγραμμα σειριακά.

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).

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 του ασθενή με το μεγαλύτερο ποσό (παράμετροι συνάρτησης: πίνακας ασθενών, πλήθος τους. Επιστρέφει τον ασθενή με το μεγαλύτερο ποσό)
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 και επιστρέφει τον ασθενή. Επιστρέφει (Boolean συνάρτηση) αν βρέθηκε ο ασθενής). Η αναζήτηση θα γίνεται με το επώνυμο και το όνομα που θα διαβάζονται (δίνονται από το χρήστη) εντός 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 4 ασθενείς & την ELPIS με 3 ασθενείς. 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:

Dose ta stoixeia toy astheni 1
Eponimo: ASPASIDOU
Onoma: ASPA
Dief8insi: NEMEAS 8
Poso pliromis: 1000

Dose ta stoixeia toy astheni 2
Eponimo: GEWRGIOU
Onoma: GEWRGIOS
Dief8insi: PILEAS 10
Poso pliromis: 200

Dose ta stoixeia toy astheni 3
Eponimo: ADREOU
Onoma: ADREAS
Dief8insi: OLGAS 10
Poso pliromis: 230

Dose ta stoixeia toy astheni 4
Eponimo: NOLAS
Onoma: NIKOS
Dief8insi: OLGAS 20
Poso pliromis: 560

Dose ta stoixeia toy astheni 1
Eponimo: TSIPAS
Onoma: VAGGELIS
Dief8insi: ALH8EIAS 100
Poso pliromis: 20

Dose ta stoixeia toy astheni 2
Eponimo: GIGAS
Onoma: GIACKOS
Dief8insi: DIAKOU 8
Poso pliromis: 123

Dose ta stoixeia toy astheni 3
Eponimo: PANTOS
Onoma: NIKOS
Dief8insi: ALMYROY 45
Poso pliromis: 89

EPWNYMO	ONOMA	DIEY8YNSH	POSO
ASPASIDOU	ASPA	NEMEAS 8	1000
GEWRGIOU	GEWRGIOS	PILEAS 10	200
ADREOU	ADREAS	OLGAS 10	230
NOLAS	NIKOS	OLGAS 20	560

EPWNYMO	ONOMA	DIEY8YNSH	POSO
TSIPAS	VAGGELIS	ALH8EIAS 100	20
GIGAS	GIACKOS	DIAKOU 8	123
PANTOS	NIKOS	ALMYROY 45	89

EPWNYMO	ONOMA	DIEY8YNSH	POSO
ASPASIDOU	ASPA	NEMEAS 8	1000
EPWNYMO	ONOMA	DIEY8YNSH	POSO
GIGAS	GIACKOS	DIAKOU 8	123

Max income GALINOS 1990

Give patient's last name NOLAS

Give patient's first name NIKOS

EPWNYMO	ONOMA	DIEY8YNSH	POSO
NOLAS	NIKOS	OLGAS 20	560

Give patient's last name TSIPAS

Give patient's first name ALEKOS

Not foundΠιέστε ένα πλήκτρο για συνέχεια. . .

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

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 **void** συνάρτηση που να υπολογίζει το συνολικό εμβαδόν όλων των πλευρών του κουτιού (πεδίο **area**). Η συνάρτηση θα ενημερώνει το αντίστοιχο πεδίο της δομής.
- Μια **void** συνάρτηση 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 τρεις συναρτήσεις να δέχονται ως παράμετρο έναν δείκτη στη δομή **box**. 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 πλήθος 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 stoixeia tou kathigiti 0
Dose onoma: Nikos
Dose eponymo: Ioannou
Dose mistho oras: 12
Dose ores ergasias: 20
Dose ta stoixeia 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
```