



8η Σειρά Ασκήσεων - Δομές και Ενώσεις

Άσκηση 8.1

Το Σφαιρικό Σύστημα Συντεταγμένων καθορίζει τη θέση ενός σημείου του τρισδιάστατου επίπεδου χώρου με τη βοήθεια δύο γωνιών και μιας απόστασης από την αρχή O . Ο μετασχηματισμός από τις Σφαιρικές Συντεταγμένες (r, θ, φ) στις Καρτεσιανές Συντεταγμένες (x, y, z) πραγματοποιείται μέσω των τύπων:

$$x = r \sin\theta \cos\varphi, \quad y = r \sin\theta \sin\varphi, \quad z = r \cos\theta$$

Να ορίσετε μια δομή με όνομα `m_coords` για τη διαχείριση υλικών σημείων μάζας m ενός στερεού σώματος. Τα μέλη της δομής είναι η μάζα και οι τρεις σφαιρικές συντεταγμένες (r, θ, φ) όπως ορίστηκαν προηγουμένως. Όλες οι τιμές είναι τύπου `double`.

Το πρόγραμμα θα πρέπει να περιλαμβάνει συναρτήσεις για:

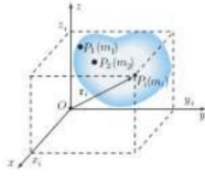
- την εισαγωγή των δεδομένων n υλικών σημείων ($n = \text{γνωστό}$) σε έναν πίνακα δομών τύπου `m_coords` (συνάρτηση `data_entry`). Να χρησιμοποιηθούν τυχαίοι αριθμοί τύπου `double` με όρια `50.0` για τη μάζα, `10.0` για την απόσταση και `2π` (σε ακτίνια) για τις γωνίες.
- την εμφάνιση των στοιχείων του πίνακα δομών σε σειρές (1 σειρά για κάθε υλικό σημείο) (συνάρτηση `data_display`).
- την εύρεση των καρτεσιανών συντεταγμένων του κέντρου μάζας του στερεού σώματος από όλα τα υλικά σημεία του πίνακα δομών, σύμφωνα με τους παρακάτω τύπους (μία συνάρτηση με όνομα `find_mass_center`):

$$x_{KM} = \frac{\sum_{i=1}^n m_i x_i}{\sum_{i=1}^n m_i}, \quad y_{KM} = \frac{\sum_{i=1}^n m_i y_i}{\sum_{i=1}^n m_i}, \quad z_{KM} = \frac{\sum_{i=1}^n m_i z_i}{\sum_{i=1}^n m_i}$$

Οι τρεις συναρτήσεις θα καλούνται διαδοχικά από τη `main()`. Μετά την κλήση της συνάρτησης `find_mass_center` το πρόγραμμα θα πρέπει να εμφανίζει τις τιμές για τα x_{KM} , y_{KM} , z_{KM} .

Άσκηση 8.2

Να ορίσετε μία δομή με όνομα `mass_point` που θα χρησιμοποιηθεί για τη διαχείριση υλικών σημείων ενός στερεού σώματος. Τα μέλη της δομής είναι η μάζα του υλικού σημείου και οι τρεις συντεταγμένες του στο χώρο, όπως φαίνεται στο σχήμα (απεικονίζονται τα υλικά σημεία P_1, P_2, \dots, P_i με μάζες αντίστοιχα m_1, m_2, \dots, m_i). Όλες οι τιμές είναι τύπου `double`.



$$I_{xOy} = \sum_i m_i z_i^2, \quad I_{yOz} = \sum_i m_i x_i^2, \quad I_{zOx} = \sum_i m_i y_i^2$$

Το πρόγραμμα θα πρέπει να περιλαμβάνει συναρτήσεις για:

1. την εισαγωγή των δεδομένων n υλικών σημείων ($n = \text{γνωστό}$) σε έναν πίνακα δομών τύπου `mass_point` (συνάρτηση `data_entry`). Να χρησιμοποιηθούν τυχαίοι αριθμοί με όρια `100.0` για τη μάζα και `30.0` για κάθε μία από τις συντεταγμένες.
2. την εμφάνιση των στοιχείων του πίνακα δομών σε σειρές (1 σειρά για κάθε υλικό σημείο) (συνάρτηση `data_display`).
3. την εύρεση των ροπών αδρανείας I_{xOy} , I_{yOz} , I_{zOx} (σε μία συνάρτηση με όνομα `find_inertia`) λαμβάνοντας υπόψη όλα τα στοιχεία του πίνακα δομών.

Οι τρεις συναρτήσεις θα καλούνται διαδοχικά από τη `main()`. Μετά την κλήση της συνάρτησης `find_inertia` το πρόγραμμα θα πρέπει να εμφανίζει τις τιμές για τα I_{xOy} , I_{yOz} , I_{zOx} .

Άσκηση 8.3

Το κέντρο βάρους (ή κεντροειδές) ενός στοιχειώδους επίπεδου σχήματος i , έχει συντεταγμένες x_i , y_i και εμβαδόν A_i . Για τον υπολογισμό του κεντροειδούς μιας διατομής (δηλαδή των \bar{x} , \bar{y}) απαιτείται ο υπολογισμός των \bar{x} , \bar{y} από όλα τα στοιχειώδη επίπεδα σχήματα που συνθέτουν τη διατομή, σύμφωνα με τους παρακάτω τύπους:

$$\bar{x} = \frac{\sum_{i=1}^n x_i A_i}{\sum_{i=1}^n A_i}, \quad \bar{y} = \frac{\sum_{i=1}^n y_i A_i}{\sum_{i=1}^n A_i}$$

Να οριστεί μια δομή (struct) με όνομα `centroid` για τη διαχείριση στοιχειωδών επίπεδων σχημάτων (τύπου `centroid`) που ανήκουν όλα στην ίδια διατομή. Τα μέλη της δομής είναι οι συντεταγμένες και το εμβαδόν, σύμφωνα με τα παραπάνω. Όλες οι τιμές είναι τύπου `double`.

Το πρόγραμμα θα πρέπει να περιλαμβάνει συναρτήσεις για:

1. την εισαγωγή των δεδομένων n στοιχειωδών επίπεδων σχημάτων ($n = \text{γνωστό}$) σε έναν πίνακα δομών τύπου `centroid` (συνάρτηση `data_entry`). Να χρησιμοποιηθούν τυχαίοι αριθμοί με όρια `50` για τις συντεταγμένες και `2000` για το εμβαδόν.
2. την εμφάνιση των στοιχείων του πίνακα δομών σε σειρές (1 σειρά για κάθε στοιχειώδες επίπεδο σχήμα) (συνάρτηση `data_display`).
3. την εύρεση των \bar{x} , \bar{y} της διατομής (μία συνάρτηση με όνομα `find_centroid`), λαμβάνοντας υπόψη όλα τα στοιχεία του πίνακα δομών.

Οι τρεις συναρτήσεις θα καλούνται διαδοχικά από τη `main()`. Μετά την κλήση της συνάρτησης `find_centroid` το πρόγραμμα θα πρέπει να εμφανίζει τις τιμές για τα \bar{x} , \bar{y} .

Άσκηση 8.4

Δεδομένα που αφορούν εργαζόμενους αποθηκεύονται σε πίνακα με όνομα `factory`, που ορίζεται σύμφωνα με τα παρακάτω :

```
struct address {
    int city_code;
    int street_code;
    int number;
};
struct person {
    int code;
    struct address addr;
    float salary;
};
struct person factory[10];
```

Να γραφεί πρόγραμμα σε γλώσσα C που :

1. θα καταχωρεί τα στοιχεία των εργαζομένων στον πίνακα `factory`.
2. θα βρίσκει και θα εκτυπώνει τα στοιχεία του εργαζόμενου με τον μεγαλύτερο μισθό με χρήση απλών μεταβλητών.
3. θα βρίσκει και θα εκτυπώνει τα στοιχεία του εργαζόμενου με τον μεγαλύτερο μισθό χρησιμοποιώντας τις παρακάτω δηλώσεις:

```
float most=factory[0].salary;
struct person *aux, *who;
```

Άσκηση 8.5

Δίνεται η δομή :

```
struct student {
    int id;
    float marks[5];
};
```

Ο πίνακας `marks` περιέχει τους βαθμούς του φοιτητή με αριθμό μητρώου `id`. Να γραφεί πρόγραμμα σε γλώσσα C που χρησιμοποιώντας τις αντίστοιχες συναρτήσεις :

1. θα καταχωρεί τα στοιχεία 10 φοιτητών σε πίνακα με στοιχεία τύπου `student` (συνάρτηση `data_entry`).

2. Θα εμφανίζει τα στοιχεία του πίνακα σε σειρές (1 σειρά για κάθε φοιτητή) (συνάρτηση `data_display`)
3. Θα βρίσκει και θα επιστρέφει στο κυρίως πρόγραμμα τους αριθμούς μητρώου των φοιτητών που έχουν περάσει όλα τα μαθήματα (συνάρτηση `pass_all`).
4. Οι συναρτήσεις θα καλούνται από τη συνάρτηση `main()`.

Άσκηση 8.6

Δίνεται η δομή:

```
struct employee {
    int code;
    float salary[12];
    int year;
};
```

Τα μέλη της δομής είναι :

- Ο κωδικός του υπαλλήλου (αποδεκτές τιμές >0)
- Οι ετήσιες αποδοχές του ανά μήνα (αποδεκτές τιμές >0)
- Το έτος πρόσληψης (αποδεκτές τιμές >2000)

Να οριστεί ένας πίνακας `N` θέσεων του τύπου `employee`.

Να γίνει εισαγωγή στοιχείων στον πίνακα με τους ελέγχους εγκυρότητας (συνιστάται η κατάλληλη χρήση της συνάρτησης `rand()`)

Να βρεθούν και να εμφανιστούν

1. Ο μέσος χρόνος προϋπηρεσίας για όλα τα στοιχεία του πίνακα.
2. Το σύνολο των ετησίων αμοιβών για όσους εργαζόμενους έχουν τουλάχιστον 5 έτη προϋπηρεσίας
3. Το σύνολο των αποδοχών για τον μήνα Μάιο
4. Το σύνολο των αποδοχών για καθέναν από τους 12 μήνες (δηλ. 12 τιμές)

Άσκηση 8.7

Να γραφεί ένα πρόγραμμα για τη διαχείριση ενός πίνακα με όνομα `products` και με στοιχεία του τύπου `product` που ορίζεται ως εξής :

```
struct description {
    int category;
    int code;
};
struct product {
    struct description d;
    int quality[12];
    int production[12];
};
```

Κάθε στοιχείο του πίνακα περιέχει τις μηνιαίες ποσότητες παραγωγής ενός προϊόντος (πίνακας `production[12]`) μαζί τον αντίστοιχο χαρακτηρισμό της παραγωγής από τον ποιοτικό έλεγχο (ένας από τους αριθμούς `1`, `2`, `3`, `4`) - (πίνακας `quality[12]`).

Το πρόγραμμα θα πρέπει να περιλαμβάνει συναρτήσεις για:

1. την εισαγωγή των δεδομένων 10 προϊόντων στον πίνακα `products` (συνάρτηση `data_entry`).
2. την εμφάνιση των στοιχείων του πίνακα `products` σε σειρές (1 σειρά για κάθε προϊόν) (συνάρτηση `data_display`).
3. την εύρεση και την εμφάνιση (στη συνάρτηση `main()`) της συνολικής ετήσιας ποσότητας παραγωγής κάθε προϊόντος με χαρακτηρισμό ποιοτικού ελέγχου `1`, `2`, `3`, `4` (δηλαδή για κάθε προϊόν πρέπει να βρεθούν και να εμφανιστούν τέσσερις συνολικές ετήσιες τιμές – μία για καθένα από τα `1`, `2`, `3`, `4`), (συνάρτηση `find_total`).
4. την εύρεση και την εμφάνιση (στη συνάρτηση `main()`) του προϊόντος που έχει το μεγαλύτερο ποσοστό % παραγωγής με χαρακτηρισμό ποιοτικού ελέγχου `4`. (συνάρτηση `max_fail`)

Οι συναρτήσεις θα καλούνται διαδοχικά από τη συνάρτηση `main()`.

Άσκηση 8.8

Οι πληροφορίες για τις μηχανές που διαθέτει μια μονάδα παραγωγής περιλαμβάνουν :

1. αριθμό μηχανής (`idnumber` - τύπου `int`)
2. περιγραφή της μηχανής (`description` – πίνακας χαρακτήρων `15` θέσεων)
3. ημερομηνία αγοράς (`purchaseDate`)
4. κόστος (`cost` - τύπου `double`)
5. το ιστορικό της (`history`), με μέλη
 - a. ποσοστό αποτυχίας (`failrate` - τύπου `double`)
 - b. αριθμός ημερών που μένει εκτός λειτουργίας (`downdays` - τύπου `int`)
 - c. ημερομηνία τελευταίας επισκευής (`lastServiced`)

Η μορφή μιας τυπικής δομής για μια μηχανή φαίνεται στο παρακάτω σχήμα :

5719	"DRILLING..."	.02	1	25	1999	4	3	21	1995	8000.0
		.failrate	.lastServiced			.downdays	.purchaseDate			.cost
			.month	.day	.year		.month	.day	.year	

machine.history.lastServiced.year έχει τιμή 1999

Να γραφεί ένα πρόγραμμα σε γλώσσα C που θα υλοποιεί τα παρακάτω :

1. Θα ορίζει τις απαραίτητες δομές (structs) :
 - **DateType** (για τις ημερομηνίες με τη μορφή που φαίνονται στο παράδειγμα)
 - **StatisticsType** (θα περιλαμβάνει τα πεδία που αναφέρονται στο ιστορικό)
 - **MachineRec** (θα περιλαμβάνει όλες τις πληροφορίες που αφορούν τη μηχανή και με τη σειρά που φαίνονται στο παράδειγμα)
2. Θα δημιουργεί στο κυρίως πρόγραμμα ένα πίνακα δομών με **n** στοιχεία του τύπου **MachineRec** (**n** = γνωστό).
3. Στη συνάρτηση **main()** :
 - Θα εισάγει δεδομένα στον πίνακα δομών με χρήση της **rand()**, εκτός από το **description**, για το οποίο η εισαγωγή θα γίνεται μέσω της **scanf_s**.
 - Θα εμφανίζει τα στοιχεία του πίνακα, ένα στοιχείο ανά γραμμή της οθόνης.
 - Θα υπολογίζει και θα εμφανίζει :
 - τη μέση τιμή κόστους των μηχανών
 - τη μέση τιμή των ημερών που οι μηχανές είναι εκτός λειτουργίας
 - το πλήθος των μηχανών που αγοράστηκαν για καθένα από τα έτη 2011-2020