



2η Σειρά Ασκήσεων - Δείκτες και Συναρτήσεις

Άσκηση 2.1

Να γράψετε μια συνάρτηση `main()` που να περιέχει κατάλληλες εντολές για την επαλήθευση αυτών των παρακάτω λειτουργιών:

- Δήλωση και αρχικοποίηση της `next` ως μεταβλητής τύπου χαρακτήρα με τιμή `'B'`.
- Δήλωση και αρχικοποίηση της `current` ως μεταβλητής τύπου χαρακτήρα με τιμή `'y'`.
- Δήλωση του `ptr` ώστε να είναι δείκτης σε αντικείμενα τύπου `char`.
- Εκχώρηση της διεύθυνσης της `current` στη μεταβλητή `ptr`.
- Αλλαγή της τιμής του αντικειμένου στο οποίο δείχνει ο `ptr` σε `'θ'`.
- Εκχώρηση της διεύθυνσης της `next` στη μεταβλητή `ptr`.
- Αλλαγή της τιμής της μεταβλητής στην οποία δείχνει ο `ptr` σε `'d'`.
- Εμφάνιση της διεύθυνσης που είναι αποθηκευμένη στην `ptr`.
- Ποιες τιμές είναι αποθηκευμένες στις `next` και `current`?

Άσκηση 2.2

Δίνονται οι παρακάτω δηλώσεις:

```
int x=0;
int y=0;
int *myPtr=NULL;
int *otherPtr=NULL;
```

Με τη βοήθεια κατάλληλων εντολών `printf`, βρείτε τις τιμές των `myPtr`, `otherPtr`, `x`, και `y`, μετά από κάθε μια από τις παρακάτω γραμμές κώδικα:

- `myPtr = &x;`
- `otherPtr = &y;`
- `*myPtr = 4;`
- `*otherPtr = *myPtr;`
- `x = 5;`
- `otherPtr = myPtr;`
- `*otherPtr = 6;`

Άσκηση 2.3

Τι πιστεύετε ότι θα γίνει αν εκτελέσετε το ακόλουθο πρόγραμμα και γιατί;

```
void main(){
    double trouble = 3.27;
    double *pt;
    *pt = 27.3;
    pt = &trouble;
}
```

Άσκηση 2.4

Τι πιστεύετε ότι θα γίνει αν εκτελέσετε το ακόλουθο πρόγραμμα και γιατί;

```
void main(){
    int i, *k;
    *k = 4;
    k = &i;
}
```

Άσκηση 2.5

Να γράψετε συνάρτηση που να δέχεται δύο ακέραιες μεταβλητές και να αντιμεταθέτει τις τιμές τους. Στη συνέχεια να γραφεί κύριο πρόγραμμα που να διαβάζει 2 ακέραιους αριθμούς, θα καλεί τη συνάρτηση και θα εμφανίζει τις τιμές τους μετά την αντιμετάθεση, Π.χ. αν έχουμε τις τιμές $a=5$ και $b=3$, μετά την εκτέλεση της συνάρτησης οι μεταβλητές αυτές θα περιέχουν τις τιμές $a=3$ και $b=5$.

Άσκηση 2.6

Σε ένα απλό αισθητήρα μιας συσκευής, ο χρόνος μετρίεται σε δευτερόλεπτα από τα μεσάνυχτα. Έτσι για παράδειγμα, όταν το ρολόι της συσκευής δείχνει 4812, η ώρα είναι 1:20:12. Γράψτε μια συνάρτηση `GetTime` με το ακόλουθο πρωτότυπο:

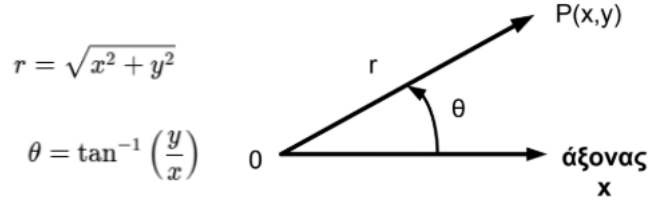
```
void getTime(long sysSecs, int *hours, int *minutes, int *seconds);
```

που να δέχεται έναν αριθμό `long` ο οποίος δηλώνει τα δευτερόλεπτα που αναφέρονται στο ρολόι της συσκευής και επιστρέφει την ώρα, τα λεπτά και τα δευτερόλεπτα που αντιστοιχούν στον κλασικό τρόπο μέτρησης του χρόνου της ημέρας.

Δοκιμάστε τη συνάρτηση, γράφοντας ένα απλό πρόγραμμα που να διαβάσει το χρόνο σε δευτερόλεπτα, να καλεί τη συνάρτηση και να εμφανίζει την ώρα, όπως στο παράδειγμα.

Άσκηση 2.7

Να γραφεί μια συνάρτηση με όνομα `convert_to_polar` που να δέχεται ως είσοδο τις καρτεσιανές συντεταγμένες ενός σημείου x, y και να επιστρέφει τις πολικές συντεταγμένες του r, θ . Οι πολικές συντεταγμένες υπολογίζονται σύμφωνα με τους τύπους:



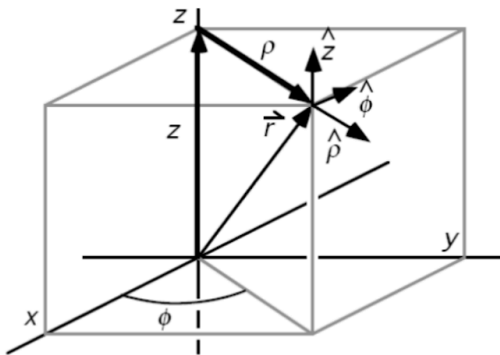
Η συνάρτηση `θ` καλείται από τη `main()`.

Άσκηση 2.8

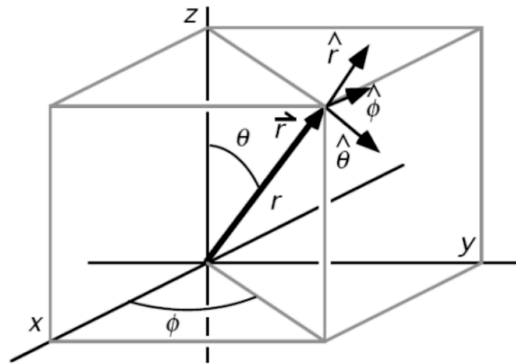
Να γράψετε δύο συναρτήσεις με ονόματα:

- `convert_to_cylindrical`
- `convert_to_spherical`

που να υπολογίζουν και να επιστρέφουν τις κυλινδρικές και σφαιρικές συντεταγμένες ενός σημείου (x, y, z) .



Κυλινδρικές



Σφαιρικές

$$\rho = \sqrt{x^2 + y^2} \quad x = \rho \cos \phi$$

$$\phi = \arctan(y, x) \quad y = \rho \sin \phi$$

$$z = z \quad z = z$$

$$r = \sqrt{x^2 + y^2 + z^2} \quad x = r \sin \theta \cos \phi$$

$$\theta = \arctan(\sqrt{x^2 + y^2}, z) \quad y = r \sin \theta \sin \phi$$

$$\phi = \arctan(y, x) \quad z = r \cos \theta$$

Στη συνέχεια, να γραφεί πρόγραμμα που:

1. Θα δημιουργεί `N` σημεία (x, y, z) στο διάστημα `[0, 100]` χρησιμοποιώντας τη συνάρτηση `rand()`. Η του `N` είναι γνωστή και θα πρέπει να οριστεί ως σταθερά.
2. Θα καλεί τις δύο συναρτήσεις για καθένα από τα `N` σημεία και θα εμφανίζει τις τιμές που επιστρέφονται από τις συναρτήσεις.

Υπόδειξη: Όλες οι τιμές να είναι τύπου `double`.

Άσκηση 2.9

Να γραφεί μια συνάρτηση σε γλώσσα C που υπολογίζει την τετραγωνική ρίζα ενός αριθμού `N` χρησιμοποιώντας την επαναληπτική μέθοδο:

$$NG = 0.5 (LG + N / LG)$$

όπου

- **NG** είναι η νέα εκτίμηση της τετραγωνικής ρίζας και
- **LG** είναι η τρέχουσα εκτίμηση.

Η διαδικασία ξεκινά με αρχική εκτίμηση $LG = 1.0$ και συνεχίζεται έως ότου η απόλυτη διαφορά μεταξύ **NG** και **LG** γίνει μικρότερη από 0.005. Όταν η συνθήκη αυτή ικανοποιηθεί, η συνάρτηση επιστρέφει την τελική εκτίμηση της ρίζας και τον αριθμό των επαναλήψεων που απαιτήθηκαν για τους υπολογισμούς αυτούς. Η συνάρτηση θα πρέπει να είναι της μορφής:

```
void sqrt_approx(double N, double *result, int *iterations)
```

Να γραφεί επίσης συνάρτηση που να δέχεται ως ορίσματα 2 πραγματικούς αριθμούς, `min`, `max` και να επιστρέφει έναν τυχαίο πραγματικό αριθμό στο διάστημα `[min, max]`.

Στη συνέχεια, να γραφεί ένα πρόγραμμα που καλεί τις παραπάνω συναρτήσεις και εμφανίζει την τετραγωνική ρίζα για 4 τυχαίους πραγματικούς αριθμούς στο διάστημα `[1.0, 200.0]`.

Άσκηση 2.10

Ο αριθμός **2022** είναι σύνθετος τετραψήφιος αριθμός. Δημιουργείται από δύο διαδοχικούς άρτιους αριθμούς σε σειρά, τους **20** και **22**, και βρίσκεται μεταξύ δύο διαδοχικών πρώτων αριθμών, των **2017** και του **2027**, και μάλιστα είναι ο μέσος όρος τους: $2022 = (2017 + 2027) / 2$.

Να γραφούν οι ακόλουθες 2 συναρτήσεις:

- συνάρτηση `void find_even_pair(int num, int *even1, int *even2)`, που να δέχεται έναν τετραψήφιο ακέραιο αριθμό και στην περίπτωση που αυτός δημιουργείται από 2 διαδοχικούς άρτιους (όπως στο παράδειγμα) να επιστρέφει αυτούς τους 2 αριθμούς
- Συνάρτηση που δέχεται έναν θετικό ακέραιο αριθμό, $k \geq 2$ και ελέγχει αν αυτός είναι πρώτος (prime) αριθμός.

Επίσης, να γραφεί κύριο πρόγραμμα που θα εμφανίζει όλους τους αριθμούς στο διάστημα `[1000, 9999]` που πληρούν και τις δύο ανωτέρω ιδιότητες.

Άσκηση 2.11

Να γραφεί συνάρτηση της μορφής:

```
int find_mk(int n, int *m, int *k);
```

που να δέχεται έναν ακέραιο αριθμό `n` και να ελέγχει αν μπορεί να γραφεί στη μορφή: $n = m^k$ (όπου `m`, `k` ακέραιοι στο διάστημα `[2, 100]`).

Η συνάρτηση θέλουμε επιστρέφει τον 1ο αποδεκτό συνδυασμό (`m`, `k`) που θα εντοπίσει. Σε περίπτωση που δεν υπάρχει αποδεκτό ζευγάρι (`m`, `k`), να επιστρέφει `-1`.

Στη συνέχεια, να γραφεί ένα πρόγραμμα που θα ζητάει από τον χρήστη έναν αριθμό, θα καλεί αυτή τη συνάρτηση, και θα εμφανίζει το αποτέλεσμα στην οθόνη.