

11.2 Άσκηση 2-Διαχείριση Αλφαριθμητικών

Η άσκηση δίνεται σε δύο εκδόσεις. Η δεύτερη, η 2β, προϋποθέτει την εκτέλεση και άλληρη κατανόηση της πρώτης έκδοσης 2α.

Άσκηση 2α

11.2.1 Περιγραφή

Να δημιουργηθεί πρόγραμμα στο οποίο θα πρέπει να δηλώνεται η μεταβλητή name όπως παρακάτω

```
char name[10] = "NIKOPY";
```

Στη συνέχεια, το πρόγραμμα θα πρέπει να ορίσει την εκτύπωση του αλφαριθμητικού name στην οθόνη με χρήση της printf σύμφωνα με τους παρακάτω πρό-
τους:

- α) δύο σε μία γραμμή με χρήση του ποσοδιοριστή %s
- β) δύο σε μία γραμμή με χρήση του ποσοδιοριστή %c
- γ) ένας χαρακτήρας ανά γραμμή
- δ) όπως παρακάτω
- NIKOY
- IKOY
- KOY
- OY
- Y
- NIK
- NIKO
- NIK
- NI
- N

Άσκηση 2β

Χρησιμοποιήστε τον πηγαίο κώδικα της πρώτης έκδοσης και τροποποιήστε τον ώστε το πρόγραμμα στη δεύτερη του έκδοση να ζητά από το χρήστη το επώνυμο του.

Εξετάστε τη συμπεριφορά του συστήματος στην περίπτωση που ο χρήστης δώσει επώνυμο με περισσότερους χαρακτήρες από τον αριθμό των στοιχείων του πίνακα name. Αιτιολογήστε τη συμπεριφορά.

Αντιμετωπίστε το πρόβλημα χωρίς να τροποποιήσετε το μέγεθος του πίνακα name.

11.2.2 Στόχος

Εξοικείωση με

- τις προτάσεις ελέγχου ροής εκτέλεσης του προγράμματος.
- την έννοια του πίνακα
- την έννοια του αλφαριθμητικού

Χρόνος εκτέλεσης: Κατά τη διάρκεια της μελέτης του κεφαλαίου 6 ή με την ολοκλήρωσή του.

11.2.3 Οδηγίες

Χρησιμοποιήστε κατάλληλα τους βρόχους επανάληψης *for*, *while*, και *do*.

Αποθήκευση αλφαριθμητικού

Θεωρώντας τον πίνακα `name[40]` το πρώτο στοιχείο του είναι το `name[0]` και το τελευταίο του το `name[39]`. Η C ορίζει ότι το όνομα του πίνακα είναι ο δείκτης στο πρώτο στοιχείο του πίνακα.

☞ **Προσοχή.** Το σύστημα αυτόματα μετά το τελευταίο χαρακτήρα του ονοματεπώνυμου σας θα τοποθετήσει τον χαρακτήρα `'\0'`. Αυτός είναι ο λόγος που πρέπει να δηλώσετε πίνακα διαστάσεως του μεγέθους του ονοματεπώνυμου σας συν ένα τουλάχιστο.

Εκτύπωση αλφαριθμητικού

Για εκτύπωση αλφαριθμητικού με την `printf`, χρησιμοποιήστε τον προσδιοριστή `%s`. Στη λίστα των ορισμάτων που ακολουθεί, η `printf` αναμένει τη διεύθυνση του αλφαριθμητικού.

Δοκιμάστε τα παρακάτω. Τι τυπώνεται σε κάθε περίπτωση;

```
printf("%s\n",name);
printf("%c\n",*name);
printf("%s\n",&name[2]);
printf("%c\n",name[2]);
```

Αιτιολογήστε το αποτέλεσμα λαμβάνοντας υπόψη ότι:

- η `printf` χρησιμοποιεί την ακολουθία `%c` για να τυπώσει ένα χαρακτήρα και ότι
- το `name` είναι δείκτης στο πρώτο στοιχείο του αλφαριθμητικού. Το `name[2]` είναι το τρίτο στοιχείο του πίνακα `name` και το `&name[2]` είναι η διεύθυνσή του.

Για το ερώτημα δ, θεωρήστε τις παρακάτω εκφράσεις

```
printf("%s\n",name);
printf("%c\n",*name);
printf("%s\n",&name[2]);
printf("%c\n",name[2]);
```

Ποιες δίνονται αιτιολογίες;

Για το ερώτημα δ, θεωρήστε τις παρακάτω εκφράσεις

A) μια έκδοση του προγράμματος που τυπώνει

```
printf("%s\n",name);
```

Ποιο είναι το αποτέλεσμα;

B) μια έκδοση του προγράμματος που τυπώνει

Ανατρέξτε στο κεφάλαιο 6 για να θυμηθείτε τι τυπώνει η `printf` με τον προσδιοριστή `%c`.

11.2.3.1 Συνάρτηση strlen

Για να χρησιμοποιήσετε τη συνάρτηση `strlen`, πρέπει να συμπεριλάβετε το αρχείο κεφαλαίου `string.h`.

```
#include <string.h>
```

Το πρωτότυπο της συνάρτησης `strlen` είναι

```
int strlen(const char *s);
```

Η συνάρτηση `strlen` επιστρέφει τον αριθμό των χαρακτήρων του αλφαριθμητικού που δίνεται ως παραμετρική.

Στο παρακάτω πρόγραμμα, ο πίνακας `name` είναι αλφαριθμητικού μήκους `name_len`.

```
name_len
```



```
printf("%s\n", name[i]);  
printf("%s\n", &name[i]);  
printf("%s\n", name[i]);  
printf("%s\n", name+i);
```

Ποιες δίνουν το επιθυμητό αποτέλεσμα;
Αντιολογήστε την απάντησή σας για κάθε περίπτωση.

Για το ερώτημα ε δώστε:

A) μια έκδοση χρησιμοποιώντας την πρόταση

```
printf("%s\n", name);
```

Ποιο είναι το μειονέκτημα της εκδοχής αυτής;

B) μια έκδοση χωρίς τη χρήση της printf.

Αναφέρετε στη βασική βιβλιοθήκη για να βρείτε κάποια συνάρτηση που θα σας βοηθήσει στην κατεβούνη αυτή.

11.3.1 Συνάρτηση strlen.

Για να χρησιμοποιηθεί η συνάρτηση strlen πρέπει να συμπεριληφθεί το αρχείο εκκεφαλίδας *string.h*.

```
#include <string.h>
```

Το πρωτότυπο της συνάρτησης strlen είναι:

```
int strlen(char[]
```

Η συνάρτηση strlen της βασικής βιβλιοθήκης επιστρέφει τον αριθμό των χαρακ-
κτηρών του αλφαριθμητικού του οποίου το όνομα θα της περάσουμε ως όρισμα.
Στο παρακάτω παράδειγμα κλήσης, η strlen υπολογίζει το μήκος του αλφα-
ριθμητικού name το οποίο και εκχωρούμε στη μεταβλητή name_len.

```
name_len = strlen(name);
```