



104

Αλγόριθμοι, Προγραμματισμός και Δομές Δεδομένων

Εργαστήριο – Κεφάλαιο 8β & 9 & 10α
Δείκτες σε πίνακα & Χαρακτήρες & Αλφαριθμητικά

Διδάσκουσα:
Σοφία Αλμπάνη



Άσκηση 8.1

8.1) Να γραφεί πρόγραμμα σε C που χρησιμοποιεί δείκτη για να τροποποιήσει στοιχεία ενός πίνακα. Δίνεται ο πίνακας **arr = {10, 20, 30, 40, 50}**. Το πρόγραμμα πρέπει:

- i. να θέσει τον δείκτη ptr να δείχνει στο πρώτο στοιχείο και να το αλλάξει σε 3,
- ii. να μετακινήσει τον δείκτη δύο θέσεις δεξιά και να αλλάξει το τρίτο στοιχείο σε 5,
- iii. να εμφανίσει το άθροισμα $arr[0] + arr[2]$.

Στόχος

Ο στόχος είναι η εξάσκηση στην αριθμητική δεικτών και στην πρόσβαση στοιχείων πίνακα μέσω pointers.



Ενδεικτική Λύση 8.1

```
#include <stdio.h>
int main()
{
    int *ptr, arr[] = {10, 20, 30, 40, 50};

    ptr = arr;
    *ptr = 3;

    ptr += 2;
    *ptr = 5;

    printf("%d\n", arr[0]+arr[2]);
}
```



Άσκηση 8.2

8.2) Να γραφεί πρόγραμμα σε C που:

- διαβάζει 31 θερμοκρασίες και τις αποθηκεύει σε πίνακα χρησιμοποιώντας **δείκτη (`arr + i`)**,
- ζητά από τον χρήστη μια τιμή-όριο **`tmp`**,
- αναζητά - με χρήση δείκτη - την **πρώτη** θερμοκρασία στον πίνακα που είναι **μικρότερη από την `tmp`**,
 - εάν βρεθεί, εμφανίζει την τιμή και τη μέρα (θέση + 1),
 - διαφορετικά, εμφανίζει μήνυμα ότι καμία θερμοκρασία δεν ήταν μικρότερη από την τιμή-όριο.

Στόχος:

Εξάσκηση σε pointer arithmetic και πρόσβαση σε πίνακα μέσω δεικτών.



Ενδεικτική Λύση 8.2

```
#include <stdio.h>
#define SIZE 31

int main()
{
    int i;
    double tmp, arr[SIZE];

    for(i = 0; i < SIZE; i++)
    {
        printf("Enter temperatures: ");
        scanf("%lf", arr+i);          /* Το arr+i είναι ισοδύναμο με &arr[i]. */
    }
    printf("Enter temperature to check: ");
    scanf("%lf", &tmp);
    for(i = 0; i < SIZE; i++)
    {
        if(*(arr+i) < tmp)
            break;
    }
    if(i == SIZE)
        printf("No temperature less than %.1f\n", tmp);
    else
        printf("The first temperature less than %.1f was %.1f in day %d\n", tmp, *(arr+i), i+1);
}
```

Άσκηση 8.3

8.3) Να γραφεί πρόγραμμα που γεμίζει έναν πίνακα 50 βαθμών χρησιμοποιώντας δείκτη και στη συνέχεια εμφανίζει τους βαθμούς σε **αντίστροφη σειρά**, μετακινώντας τον δείκτη από το τέλος προς την αρχή.

Βήματα:

- Μέσω βρόχου while, να διαβάζει βαθμούς, αποθηκεύοντάς τους στις θέσεις του πίνακα όπου δείχνει ο δείκτης: **scanf("%f", ptr);** και να προχωρά στη επόμενη θέση με **ptr++** μέχρι το τέλος του πίνακα.
- Μετά την ολοκλήρωση της εισαγωγής, να μετακινεί τον δείκτη μία θέση πίσω ώστε να δείχνει στο τελευταίο έγκυρο στοιχείο: **ptr--;**
- Να εμφανίζει τους βαθμούς σε αντίστροφη σειρά, ξεκινώντας από το τέλος προς την αρχή του πίνακα, χρησιμοποιώντας ξανά **ptr--** σε δεύτερο βρόχο while.

Στόχος:

Δαχείριση πινάκων αποκλειστικά μέσω δεικτών, αριθμητική δεικτών (μετακίνηση δεικτών μέσα στον πίνακα), ανάγνωσης και εκτύπωσης δεδομένων χωρίς χρήση συμβατικής δεικτοδότησης (**arr[i]**).



Ενδεικτική Λύση 8.3

```
#include <stdio.h> /

#define SIZE 50

int main()
{
    float *ptr, arr[SIZE];

    ptr = arr;
    while(ptr < arr+SIZE)
    {
        printf("Enter grade: ");
        scanf("%f", ptr);
        ptr++;
    }
    ptr--;

    while(ptr >= arr)
    {
        printf("%f\n", *ptr);
        ptr--;
    }
}
```

Άσκηση 9.1

9.1) Να γραφεί πρόγραμμα σε C το οποίο:

- Διαβάζει **τρεις διαδοχικούς χαρακτήρες** από το πληκτρολόγιο χωρίς να αγνοεί κενά, αλλαγές γραμμής ή άλλους ειδικούς χαρακτήρες.
- Ελέγχει αν οι τρεις χαρακτήρες είναι **διαδοχικοί στον ASCII κώδικα** (δηλαδή αν ο δεύτερος είναι ο αμέσως επόμενος του πρώτου και ο τρίτος ο αμέσως επόμενος του δεύτερου).
- Αν οι χαρακτήρες είναι διαδοχικοί, το πρόγραμμα εμφανίζει το μήνυμα: «Consecutive». ιαφορετικά εμφανίζει το μήνυμα: «Not Consecutive».

Στόχος

Εξάσκηση στη χρήση χαρακτήρων στη C, στη λήψη εισόδου μέσω της scanf χωρίς παράλειψη χαρακτήρων, καθώς και στη σύγκριση τιμών ASCII.



Ενδεικτική Λύση 9.1

```
#include <stdio.h>
int main()
{
    char ch1, ch2, ch3;

    printf("Enter characters: ");
    scanf("%c%c%c", &ch1, &ch2, &ch3);

    if((ch1+1 == ch2) && (ch2+1 == ch3))
        printf("Consecutive\n");
    else
        printf("Not Consecutive\n");
}
```

Άσκηση 9.2

9.2) Να γραφεί πρόγραμμα που διαβάζει έως 10 χαρακτήρες με `while((ch = getchar()) != '\n' && ch != EOF)`

- Κάθε χαρακτήρας μπορεί να είναι **ψηφίο** ή **κεφαλαίο γράμμα**.
- Τα γράμματα να μετατρέπονται στο αντίστοιχο ψηφίο τηλεφωνικού πληκτρολογίου, ενώ τα ψηφία να εμφανίζονται όπως είναι.
- Αν δοθεί μη έγκυρος χαρακτήρας, να εμφανίζεται μήνυμα λάθους και το πρόγραμμα να τερματίζει.
- Αν εισαχθούν 10 έγκυροι χαρακτήρες, να εμφανίζεται το μήνυμα «*Number completed*».

Στόχος

Να εξασκηθεί ο φοιτητής στη χρήση πινάκων αντιστοίχισης, στην επεξεργασία χαρακτήρων με `getchar()`, και στον έλεγχο εγκυρότητας εισόδου.

Βοήθεια:

Θα ετοιμάσουμε έναν πίνακα για να κάνουμε άμεσα τις αντιστοιχίσεις.

```
arr[26] = {'2', '2', '2', '3', '3', '3', '4', '4', '4', '5', '5', '5',  
'6', '6', '6', '7', '7', '7', '7', '8', '8', '8', '9', '9', '9', '9'};
```

- ABC → 2
- DEF → 3
- GHI → 4
- JKL → 5
- MNO → 6
- PQRS → 7
- TUV → 8
- WXYZ → 9



Ενδεικτική Λύση 9.2

```
#include <stdio.h> int main()
{
    int ch, dig, arr[26] = {'2', '2', '2', '3', '3', '3', '4', '4', '4', '5', '5', '5', '6', '6', '6', '7', '7', '7', '7', '8', '8', '8',
'9', '9', '9', '9'}; /* Ετοιμάσαμε έναν πίνακα για να κάνουμε άμεσα τις αντιστοιχίσεις. */
    dig = 0;
    printf("Enter digit or letter: ");
    while((ch = getchar()) != '\n' && ch != EOF)
    {
        if(ch >= 'A' && ch <= 'Z')
        {
            dig++;
            putchar(arr[ch-'A']); /* Ανάλογα με την τιμή του ch εμφανίζεται το αντίστοιχο ψηφίο. Για παράδειγμα, αν ο
χρήστης εισάγει 'I' θα εμφανιστεί το arr[73-65] = 4. */
        }
        else if(ch >= '0' && ch <= '9')
        {
            dig++;
            putchar(ch); /* Τα ψηφία εμφανίζονται όπως είναι. */
        }
        else
        {
            printf("Error: Not valid character\n");
            return 0;
        }
        if(dig == 10)
            break;
    }
    if(dig == 10)
        printf("\nNumber completed\n");
}
```

Άσκηση 10.1

10.1) Να γραφεί πρόγραμμα σε C που δημιουργεί ένα string αποτελούμενο από:

- όλα τα πεζά γράμματα της αγγλικής αλφαβήτου (a–z) και
- αμέσως μετά όλα τα κεφαλαία γράμματα (A–Z), χωρίς κενά ή άλλα σύμβολα.

Το string να αποθηκεύεται σε **πίνακα χαρακτήρων** κατάλληλου μεγέθους και στο τέλος να τερματίζεται με τον **χαρακτήρα '\0'**. Τέλος, να εμφανιστεί ολόκληρο το string στην οθόνη.

Στόχος

Εξάσκηση στη δημιουργία και συμπλήρωση strings με χρήση πινάκων χαρακτήρων, στη χρήση επαναληπτικής δομής για τη δημιουργία ακολουθιών χαρακτήρων με βάση τον ASCII κώδικα, στη σωστή τοποθέτηση του τερματικού χαρακτήρα '\0', και στην εκτύπωση ενός string με τη printf.



Ενδεικτική Λύση 10.1

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    char str[53];
```

```
    int i;
```

```
    for(i = 0; i < 26; i++)
```

```
    {
```

```
        str[i] = 'a'+i;
```

```
        str[26+i] = 'A'+i;
```

```
    }
```

```
    str[52] = '\\0'; /* Στο τέλος, προσθέτουμε τον τερματικό χαρακτήρα. */
```

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

```
}
```

Άσκηση 10.2

10.2) Να γραφεί πρόγραμμα σε C το οποίο διαβάζει μια συμβολοσειρά από το πληκτρολόγιο και ελέγχει αν αυτή αποτελεί **παλίνδρομο**, δηλαδή αν διαβάζεται το ίδιο από την αρχή προς το τέλος και από το τέλος προς την αρχή.

- Η είσοδος να αποθηκεύεται σε πίνακα χαρακτήρων, και να υπολογίζεται το μήκος της.
- Ο έλεγχος να γίνεται συγκρίνοντας τους συμμετρικούς χαρακτήρες του string (πρώτος-τελευταίος, δεύτερος-προτελευταίος κ.ο.κ.).
- Αν εντοπιστεί διαφορά χαρακτήρων, το πρόγραμμα πρέπει να εμφανίζει ότι η συμβολοσειρά **δεν είναι παλίνδρομο**, διαφορετικά να εμφανίζει ότι **είναι**.

Στόχος

Εξάσκηση στη χρήση και διαχείριση συμβολοσειρών, στον υπολογισμό μήκους string, στη σύγκριση χαρακτήρων σε συμμετρικές θέσεις, στη χρήση βρόχου που ελέγχει μόνο το πρώτο μισό της λέξης, και στη βασική λογική αναγνώρισης παλίνδρομων λέξεων.



Ενδεικτική Λύση 10.2

```
#include <stdio.h>

int main()
{
    char str[100], ch;
    int i, len = 0, diff = 0;

    printf("Enter text: ");    /* Διαβάζουμε χαρακτήρα-χαρακτήρα μέχρι Enter ή EOF */
    while((ch = getchar()) != '\n' && ch != EOF)
    {
        if(len < 99)          /* αποφυγή υπερχείλισης */
            str[len++] = ch;
    }

    str[len] = '\0';          /* τερματισμός string */

    /* Έλεγχος για παλίνδρομο */
    for(i = 0; i < len / 2; i++)
    {
        if(str[i] != str[len - 1 - i])
        {
            diff = 1;
            break;
        }
    }

    if(diff)
        printf("%s is not a palindrome\n", str);
    else
        printf("%s is a palindrome\n", str);
}
```



Άσκηση 10.3

10.3) Να γραφεί πρόγραμμα σε C το οποίο δημιουργεί ένα string μέσα σε έναν πίνακα με scanf. Το πρόγραμμα στη συνέχεια πρέπει να συγκρίνει το περιεχόμενο του πίνακα με τη συμβολοσειρά "test" χρησιμοποιώντας τη συνάρτηση strcmp. Αν τα δύο strings είναι ίδια, να εμφανίζεται το μήνυμα **"Same"**, διαφορετικά να εμφανίζεται **"Not same"**.

Στόχος

Εξάσκηση στη χρήση της strcmp() για σύγκριση strings, στη διάκριση μεταξύ αντιστοίχισης χαρακτήρων και σωστής διαχείρισης του τερματικού χαρακτήρα '\0', και στη βασική λογική ελέγχου ισότητας συμβολοσειρών



Ενδεικτική Λύση 10.3

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

int main()
{
    char str[50];

    printf("Enter word: ");
    scanf("%49s", str);    /* ασφαλές: διαβάζει έως 49 χαρακτήρες + '\0' */

    if(strcmp(str, "test") == 0)
        printf("One\n");
    else
        printf("Two\n");
}
```

Συνοπτικά

Κατηγορία	Συντακτικό	Περιγραφή
Δήλωση string	<code>char str[] = "test";</code>	Δημιουργεί string με αυτόματο <code>\0</code> .
Σύγκριση strings	<code>strcmp(a, b) == 0</code>	Σωστός τρόπος σύγκρισης strings.
Ανάγνωση χαρακτήρα	<code>ch = getchar();</code>	Διαβάζει 1 χαρακτήρα από την είσοδο.
Loop μέχρι Enter	<code>while((ch = getchar()) != '\n')</code>	Διαβάζει πολλούς χαρακτήρες μέχρι Enter.
Πρόσβαση μέσω pointer	<code>p[i]</code> ή <code>*(p+i)</code>	Ισοδύναμη πρόσβαση σε στοιχεία.
Pointer arithmetic	<code>p++</code>	Μετακινεί δείκτη στο επόμενο στοιχείο του τύπου.
ASCII έλεγχος	<code>ch >= '0' && ch <= '9'</code>	Έλεγχος ψηφίου ή γράμματος με βάση ASCII.