



Προγραμματισμός I

ΠΙΝΑΚΕΣ - ΑΛΦΑΡΙΘΜΗΤΙΚΑ

Τμήμα Πληροφορικής & Επικοινωνιών
Δρ. Θεόδωρος Γ. Λάντζος
<http://www.teiser.gr/icd/staff/lantzog>
lantzog@teiser.gr

Μονοδιάστατοι Πίνακες (tables)

- Μια συλλογή μεταβλητών ίδιου τύπου οι οποίες είναι αποθηκευμένες σε διαδοχικές θέσεις μνήμης
- Τύπους Πινάκων τόσους όσους και οι τύποι δεδομένων
- Κύκλος ζωής όπως και οι μεταβλητές
- Δήλωση κανονικά πριν την χρήση
- Φορμαλισμό
 - Τύπος_δεδομένων όνομα_πίνακα[μέγεθος]
- Για την δήλωση έχουμε τρία στοιχεία
 - α. τύπος
 - β. όνομα
 - γ. μέγεθος

```
float temp[31];
```

Αναφορά σε στοιχεία πίνακα

- Η αναφορά σε στοιχεία του πίνακα γίνεται σε συνδυασμό με το όνομα του πίνακα και με την χρήση ενός δείκτη ο οποίος εκφράζει τη σειρά του στοιχείου μέσα στον πίνακα
 - `float temp[31];`
 - `temp[0]` -> αναφερόμαστε στο πρώτο στοιχείο του πίνακα
 - `temp[1]` -> αναφερόμαστε στο δεύτερο στοιχείο του πίνακα
 - `temp[30]` -> αναφερόμαστε στο τελευταίο στοιχείο του πίνακα

Αρχικές τιμές σε πίνακα

- Μπορούμε να αναθέσουμε αρχικές τιμές στον πίνακα όπως και στις μεταβλητές
- Που τις αναθέτουμε ; -> Ακριβώς στην δήλωση όπως με τις μεταβλητές
- Για αρχικές τιμές σε πίνακα χρειαζόμαστε τους τελεστές ({}) και (,)
- πχ. `float temp[5] = {1,2,-4.2,6,8};`
- πχ. `float temp[5] = {1,2,-4.2};`

Ανάγνωση τιμών στοιχείου πίνακα

■ Για τις μεταβλητές είχαμε

```
main()
{ int var1;
  printf("Give a value :");
  scanf("%d",&var1);
  printf("\nYou gave :%d",var1);
}
```

Ανάγνωση – Εμφάνιση μεταβλητής σε πρόγραμμα C

■ Για όλα τα στοιχεία του πίνακα

```
main()
{ int i,var[20];
  for(i=0;i<20;i++)
  {
    printf("Give a value :");
    scanf("%d",&var[i]);
  }
  for(i=0;i<20;i++)
  {
    printf("\n The %d element is :%d",i,var[i]);
  }
  getch();
}
```

■ Για ένα στοιχείο πίνακα

```
main()
{ int var[20];
  printf("Give a value :");
  scanf("%d",&var[0]);
  printf("\nYou gave :%d",var[0]);
}
```

Ανάγνωση – Εμφάνιση του πρώτου στοιχείου Πίνακα σε πρόγραμμα C

Πολυπλοκότητες στους Πίνακες

- Όταν αποδίδονται αρχικές τιμές μπορεί να παραληφθεί το μέγεθος του πίνακα
 - `char pin[] = {'a','b','c','d'};`
- Οι δείκτες των στοιχείων ενός πίνακα ξεκινούν από το 0 και όχι από το 1
- Ο δείκτης έχει διαφορετική σημασία στην δήλωση και διαφορετική στην αναφορά. Στην δήλωση καθορίζει μέγεθος ενώ στην αναφορά προσδιορίζει το στοιχείο αναφοράς
- Το μέγεθος σε bytes ενός πίνακα μπορεί να προσδιοριστεί χρησιμοποιώντας τον τελεστή `sizeof`
- Η χρήση της `sizeof` στο όνομα του πίνακα επιστρέφει το συνολικό μέγεθος ενώ στο όνομα του πίνακα με δείκτη ενός στοιχείου επιστρέφει το μέγεθος του στοιχείου

Πολυδιάστατοι Πίνακες

- Πίνακας μιας διάστασης των οποίων τα στοιχεία είναι επίσης πίνακας
 - Π.χ. `int array[4][12];`
- Δήλωση κανονικά με φορμαλισμό
Τύπος_Δεδομένων όνομα_πίνακα[πλήθος γραμμών][πλήθος στηλών];
- Ο πολυδιάστατος πίνακας αποθηκεύεται στη μνήμη ως μία ακολουθία στοιχείων μίας διάστασης
- Για την δήλωση έχουμε τρία στοιχεία
 - α. τύπος
 - β. όνομα
 - γ. μέγεθος σε πλήθος γραμμών και στηλών
- Π.χ. `int array[10][8];`

Αναφορά σε στοιχεία του πίνακα

- Χρήση δεικτών τόσοι όσοι είναι αναγκαίοι για το μέγεθος των διαστάσεων
- `temp[1][3]` αναφέρεται στο τέταρτο στοιχείο της δεύτερης γραμμής
- Οι πολυδιάστατοι πίνακες αποθηκεύονται κατά γραμμές, που σημαίνει ότι ο τελευταίος δείκτης θέσης μεταβάλλεται ταχύτερα κατά την προσπέλαση των στοιχείων
- `int temp[2][3] = { {0,1,2}, {3,4,5} }; Σχ. 6.1`
- Δυνατότητα ύπαρξης απροσδιόριστων τιμών εντός πίνακα
- Προσοχή στην αρχικοποίηση του πίνακα προς αποφυγή προβλημάτων υπολογισμών.

Αλφαριθμητικό

- Αλφαριθμητικό ή συμβολοσειρά (string) είναι ένας πίνακας χαρακτήρων που τερματίζει με το μηδενικό (null) χαρακτήρα.
- Ο μηδενικός χαρακτήρας έχει τιμή ASCII με κωδικό 0 και αναπαρίσταται από την ακολουθία διαφυγής \0
- Δήλωση αλφαριθμητικού όπως και τους μονοδιάστατους πίνακες με την διαφορά ότι είναι μόνο τύπου char
 - char όνομα[μήκος];
- Διακρίνουμε τον τύπο που μπορεί να είναι μόνου τύπου char, το όνομα του αλφαριθμητικού και το μήκος που περιγράφει το πλήθος σε χαρακτήρες

Αλφαριθμητικές Σταθερές

- Η δυνατότητα ανάθεσης μιας αλφαριθμητικής τιμής σε μια μεταβλητή η οποία θα παραμείνει η ίδια για όλο το πρόγραμμα
- Η τιμή μιας αλφαριθμητικής σταθεράς περικλείεται σε διπλά εισαγωγικά “ “
- Ο μεταγλωττιστής θέτει αυτόματα στο τέλος του αλφαριθμητικού τον μηδενικό χαρακτήρα.
- Π.χ Η αλφαριθμητική Σταθερά “HELLO” απαιτεί 6 bytes όπως ακολούθως
‘H’ ‘E’ ‘L’ ‘L’ ‘O’ ‘\0’
- Η διαφορά ανάμεσα στην σταθερά ‘A’ και την αλφαριθμητική σταθερά “A” είναι ότι η πρώτη απαιτεί 1 byte για την αποθήκευση, ενώ η δεύτερη απαιτεί 2 byte ένα για τον χαρακτήρα ‘A’ και ένα για τον μηδενικό χαρακτήρα

Αρχικές τιμές σε αλφαριθμητικά

- Όπως δίνουμε αρχικές τιμές και σε μονοδιάστατους πίνακες με την διαφορά ότι οι τιμές θα είναι χαρακτήρας

Π.χ `char isbn[]={‘0’,‘-’,‘4’,‘9’,‘-’,‘7’,‘4’,‘3’,‘-’,‘3’,‘\0’};`

Μπορούμε και εναλλακτικά να περιγράψουμε

```
char isbn[]="0-49-743-3";
```

! Προσοχή στην πρώτη περίπτωση όπου έχουμε λίστα στοιχείων να συμπεριλάβουμε ως τελευταία τιμή το `null ‘\0’`

Είσοδος αλφαριθμητικού

- Για είσοδο αλφαριθμητικού έχουμε την μορφοποιούμενη συνάρτηση `scanf` και τον προσδιοριστή `%s`
 - `scanf("%s",isbn);`
- Προσοχή! Δεν χρειάζεται ο τελεστής `&` πριν από το όνομα στη μεταβλητή `isbn` διότι το όνομα του αλφαριθμητικού αναπαριστά την διεύθυνση του πρώτου στοιχείου
- Εναλλακτικά μπορούμε να χρησιμοποιήσουμε και την συνάρτηση `gets` με μορφή
 - `gets(όνομα πίνακα χαρακτήρων);`
- Η συνάρτηση `gets` διαβάζει χαρακτήρες από το πληκτρολόγιο έως ότου πατηθεί το `ENTER`
- Όταν το μέγεθος του αλφαριθμητικού είναι `N` τότε μπορεί να αποθηκεύσει μέχρι `N-1` χαρακτήρες
- Οι συναρτήσεις εισόδου `scanf` και `gets` δεν εκτελούν έλεγχο ορίων έτσι όταν κλιθούν να διαβάσουν ένα αλφαριθμητικό μεγαλύτερο από το μέγεθος του πίνακα τότε αυξάνεται το μέγεθος του πίνακα.

Εκτύπωση Αλφαριθμητικού

- Αλφαριθμητική Σταθερά με χρήση της printf
 - printf("HELLO");
- Για αλφαριθμητικά έχουμε την χρήση της printf μαζί με τον προσδιοριστή %s
 - printf("The ISBN code is : %s",isbn);
- Εναλλακτικά μπορούμε να χρησιμοποιήσουμε την συνάρτηση puts
 - puts(όνομα πίνακα χαρακτήρων);
- Η Συνάρτηση puts δεν χρειάζεται δείκτη δίπλα στο όνομα του πίνακα καθώς επίσης παρουσιάζει και την αδυναμία μορφοποίησης της εξόδου

Διαχείριση Αλφαριθμητικών

- Η γλώσσα προγραμματισμού C διαθέτει ένα σύνολο συναρτήσεων αποκλειστικά για επεξεργασία αλφαριθμητικών οι οποίες ορίζονται στο αρχείο κεφαλίδας <string.h>

| ΛΕΙΤΟΥΡΓΙΑ | ΟΛΟΙ ΟΙ ΧΑΡΑΚΤΗΡΕΣ | ΟΙ Ν ΠΡΩΤΟΙ ΧΑΡΑΚΤΗΡΕΣ |
|----------------------------|--------------------|------------------------|
| Εύρεση μήκους string | strlen() | |
| Αντιγραφή string | strcpy() | strncpy() |
| Συνένωση 2 strings | strcat() | strncat() |
| Σύγκριση 2 strings | strcmp() | strncmp() |
| Εύρεση χαρακτήρα σε string | strchr() | strrchr() |
| εύρεση string σε string | strstr() | |

Η Συνάρτηση μήκους Αλφαριθμητικού

- `strlen()`
- Επιστρέφει τον αριθμό χαρακτήρων του αλφαριθμητικού χωρίς να περιλαμβάνει το μηδενικό χαρακτήρα
- Δέχεται σαν παράμετρο το όνομα του αλφαριθμητικού
 - Πχ `char name[12] = "abcd"`
`printf*("%d", strlen(name)); -> 4`

Η συνάρτηση αντιγραφής αλφαριθμητικού

- `strcpy(αλφαριθμητικό No1, αλφαριθμητικό No2)`
- Αντιγράφει ένα αλφαριθμητικό σε ένα άλλο
- Δέχεται δύο ορίσματα όπου το No1 είναι ο προορισμός και το No2 είναι η πηγή
- Εξέλιξη της `strcpy()` είναι η `strncpy()` η οποία έχει την μορφή
- `strncpy(αλφαριθμητικό No1, αλφαριθμητικό No2, N)`
- Αντιγράφει τους N πρώτους χαρακτήρες από το No2 στο N1.

Η συνάρτηση συνένωσης αλφαριθμητικών

- `strcat`(αλφαριθμητικό No1, αλφαριθμητικό No2)
- Προσθέτει στο τέλος του No1 αλφαριθμητικού το No2
- `strncat`(αλφαριθμητικό No1, αλφαριθμητικό No2, N)
- Προσθέτει στο τέλος του No1 αλφαριθμητικού τους N χαρακτήρες από το No2 Αλφαριθμητικό

Η Συνάρτηση σύγκρισης αλφαριθμητικών

- `strcmp(name1, name2)`
- Συγκρίνει τα δύο αλφαριθμητικά και επιστρέφει την τιμή 0 εάν είναι όμοια
- `strcmp(name1, name2, N)`
- Συγκρίνει τους πρώτους N χαρακτήρες από τα δύο αλφαριθμητικά