



Ηλεκτρονικοί Υπολογιστές

Ενότητα 9: Πίνακες στη C++

..

Ζαχαρούλα Ανδρεοπούλου

Τμήμα Δασολογίας & Φυσικού Περιβάλλοντος

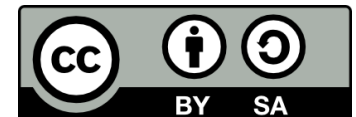


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.





ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΧΤΑ
ΑΚΑΔΗΜΑΪΚΑ
ΜΑΘΗΜΑΤΑ



Πίνακες στη C++

Περιεχόμενα ενότητας

1. Γενικά
2. Ορισμός πινάκων
3. Καθορισμός στοιχείων πίνακα
4. Εισαγωγή τιμών σε πίνακα
5. Χρήση τιμών πίνακα
6. Αναζήτηση μέσα σε έναν πίνακα
7. Ταξινόμηση ενός πίνακα
8. Εκτύπωση πίνακα
9. Καθορισμός ανώτατων ορίων πίνακα
10. Αρχικές τιμές πίνακα



Γενικά

- Ένας πίνακας (array) χρησιμοποιείται για την επεξεργασία ενός συνόλου δεδομένων, του ίδιου τύπου, όπως μία λίστα τηλεφωνικών αριθμών ή μία λίστα ονομάτων.
- Οι πίνακες μπορεί να είναι μιας διαστάσεως (διανύσματα), δύο, ή και περισσότερων διαστάσεων.



Ορισμός πινάκων 1/2

- Για να δηλώσουμε έναν πίνακα, γράφουμε στην αρχή του προγράμματος το όνομά του και το πλήθος των αριθμών (δεδομένων) που μπορεί να έχει. Επίσης, πρέπει να καθορίσουμε τον τύπο των αριθμών που θα περιέχει ακέραιοι, πραγματικοί, κλπ.
- **Παράδειγμα**

```
void main ( )  
  
{  
  
int a[20], k[30];  
  
float b[30], c[5] [10];  
  
.....}
```



Ορισμός πινάκων 2/2

- Ο πίνακας a είναι μιας διαστάσεως 20 θέσεων και μπορεί να πάρει μέχρι 20 ακέραιους αριθμούς.
- Ο πίνακας c είναι δύο διαστάσεων και μπορεί να πάρει μέχρι $5 \times 10 = 50$ πραγματικούς ή ακέραιους αριθμούς.
- Ο πίνακας b είναι 30 θέσεων, ο πίνακας k είναι 30 θέσεων και όλοι του οι αριθμοί πρέπει να είναι ακέραιοι.



Καθορισμός στοιχείων πίνακα 1/2

- Κάθε στοιχείο καθορίζεται από τη θέση που κατέχει μέσα στον πίνακα.
- Για πίνακες μίας διάστασης, το πρώτο στοιχείο καθορίζεται από τη θέση 0 (δείκτης μηδέν), το δεύτερο στοιχείο καθορίζεται από τη θέση 1 και το τελευταίο στοιχείο καθορίζεται από τη θέση $n-1$.
- **Παράδειγμα πίνακα μίας διάστασης**

$c[0]$ -> καθορίζει το πρώτο στοιχείο του πίνακα c

$c[7]$ -> καθορίζει το όγδοο στοιχείο του πίνακα c



Καθορισμός στοιχείων πίνακα 2/2

- Παράδειγμα πίνακα δύο διαστάσεων

$a[0][0]$ -> καθορίζει το στοιχείο του πίνακα a που βρίσκεται στη θέση $(1, 1)$ του πίνακα

$a[3][5]$ -> καθορίζει το στοιχείο του πίνακα a που βρίσκεται στη θέση $(4, 6)$ του πίνακα

- Κάθε στοιχείο του πίνακα καθορίζεται από τον δείκτη i , ο οποίος παίρνει τιμές από 0 έως $n-1$.

$$A[i] \quad | \quad 0 \leq i \leq n-1$$



Εισαγωγή τιμών σε πίνακα 1/2

Υπάρχουν δύο τρόποι για να δώσουμε τιμές σε έναν πίνακα:

1) Μέσα στο πρόγραμμα

```
void main ( )
{
    float x1[10], a[10] [15];
    int i;
    .....;
x1[0] = 50;
x1[3] = -37;
a[0] [1] = 12.5;
    .....;
    for (i=0; i<10; i++)
        x[i] = 1;
    .....;}
```



Εισαγωγή τιμών σε πίνακα 2/2

Υπάρχουν δύο τρόποι για να δώσουμε τιμές σε έναν πίνακα:

1) Μέσα ανάγνωση αριθμών από το πληκτρολόγιο

```
void main ( )                               for (j=0; j<15; i++)
{                                             cin >> c[i]
[j];
int x1[10], c[10] [15];                       .....;}
float a[10] [15];
int l, j;
.....;
cin >> c[2] [3]
    >> x1[0];
.....;
for (i=0; i<10; i++)
```



Χρήση τιμών πίνακα 1/2

- Κάθε αριθμός που υπάρχει σε έναν πίνακα μπορεί να χρησιμοποιηθεί σε πράξεις, συγκρίσεις, κλπ, αρκεί φυσικά να καθορίσουμε τη θέση του μέσα στον πίνακα.
- Η μνήμη του Η/Υ αποτελείται από μία λίστα αριθμημένων θέσεων που ονομάζονται bytes. Ο αριθμός ενός byte είναι γνωστός ως η διεύθυνση (address) του byte.
- Μια απλή μεταβλητή υλοποιείται ως ένα τμήμα της μνήμης, το οποίο αποτελείται από έναν αριθμό συνεχόμενων bytes.
- Ο αριθμός των bytes καθορίζεται από τον τύπο της μεταβλητής.



Χρήση τιμών πίνακα 2/2

- Μία απλή μεταβλητή περιγράφεται στη μνήμη από δύο τμήματα πληροφορίας:
 - Τη διεύθυνση στη μνήμη (η οποία δίνει τη θέση του πρώτου byte για αυτήν τη μεταβλητή) και
 - Τον τύπο της μεταβλητής (ο οποίος λέει πόσα byte μνήμης απαιτεί η μεταβλητή).
- Όταν το πρόγραμμα αποθηκεύει μία τιμή στη μεταβλητή, η τιμή τοποθετείται σε εκείνα τα bytes της μνήμης που έχουν εκχωρηθεί σε αυτήν τη μεταβλητή.
- Οι θέσεις των διαφόρων στοιχείων του πίνακα τοποθετούνται η μία δίπλα στην άλλη μέσα στη μνήμη.



Αναζήτηση μέσα σε έναν πίνακα 1/2

- Μια συνηθισμένη εργασία στον προγραμματισμό είναι η αναζήτηση μίας τιμής μέσα σε έναν πίνακα.
- Για αυτόν τον σκοπό, χρησιμοποιούμε τον **αλγόριθμο ακολουθιακής αναζήτησης (sequential research)**.
- Για να μάθουμε τον δείκτη του στοιχείου που έχει την τιμή που αναζητάμε, χρησιμοποιούμε την συνάρτηση **search**.
- Εάν η επιθυμητή τιμή δεν βρίσκεται μέσα στον πίνακα, η συνάρτηση **search** επιστρέφει την τιμή **-1**.



Αναζήτηση μέσα σε έναν πίνακα 2/2

- Η συνάρτηση `search` χρησιμοποιεί έναν βρόχο `while` ώστε να ελέγξει τα στοιχεία του πίνακα ένα προς ένα και να διαπιστώσει αν κάποιο από αυτά ισούται με την επιθυμητή τιμή.
- Η μεταβλητή **`found`** χρησιμοποιείται ως ενδείκτης ή σημαία (flag) για να καταγράψει εάν έχει βρεθεί ή όχι το επιθυμητό στοιχείο.
- Εάν το επιθυμητό στοιχείο έχει βρεθεί μέσα στον πίνακα, τότε η μεταβλητή `found` τίθεται στην τιμή `true`, η οποία τερματίζει το βρόχο `while`.



Ταξινόμηση ενός πίνακα

- Μία από τις πιο ευρέως χρησιμοποιούμενες εργασίες προγραμματισμού είναι η ταξινόμηση μιας λίστας τιμών, όπως μιας λίστας δεικτών πωλήσεων, η οποία πρέπει να ταξινομηθεί κατά αύξουσα ή κατά φθίνουσα σειρά ή μία λίστα ονομάτων, οι οποίες πρέπει να ταξινομηθούν κατά αλφαβητική σειρά.
- Για αυτόν τον σκοπό, χρησιμοποιούμε τη συνάρτηση **sort**.



Εκτύπωση πίνακα 1/2

- Η εκτύπωση ενός μέρους του πίνακα ή ολόκληρου του πίνακα γίνεται με τη χρήση των αντίστοιχων δεικτών, οι οποίοι καθορίζουν τη θέση των στοιχείων μέσα στον πίνακα.
- **Παράδειγμα**

```
void main ( )  
  
{float b[10];  
  
int i, n=10;  
  
.....;  
  
for (i=0; i<10; i++)  
  
cout << '\n' << b[i];}
```



Εκτύπωση πίνακα 2/2

- Κάθε στοιχείο του πίνακα θα τυπωθεί σε νέα γραμμή γιατί στην cout υπάρχει ο χαρακτήρας διαφυγής. Εάν θέλουμε να τυπώσουμε πολλά δεδομένα στην ίδια γραμμή, θα παραλείψουμε τον χαρακτήρα διαφυγής.
- Μετά την εντολή cout, έχουμε βάλει κενό ώστε οι αριθμοί που τυπώνονται να έχουν κενό μεταξύ τους.
- Εάν οι αριθμοί δεν χωράνε στη γραμμή, η εκτύπωση θα συνεχιστεί στην επόμενη γραμμή.



Καθορισμός ανώτατων ορίων πίνακα

- Για να καθορίσουμε τα ανώτατα όρια ενός πίνακα, χρησιμοποιούμε την εντολή **define**.
- **Παράδειγμα**

```
#include <stdio.h>
#define LIM 10
#define NR 10
#define NC 20
void main ( )
{int array (LIM);
    float x[NR] [NC];
    .....;}
```



Αρχικές τιμές πίνακα 1/3

- Δεν μπορούμε να δώσουμε αρχικές τιμές σε έναν τοπικό πίνακα.
- Ο πίνακας θα πρέπει να οριστεί ως εξωτερικός της συνάρτησης στην οποία ανήκει.
- Ο πίνακας δηλώνεται πριν την κύρια συνάρτηση και πριν κάθε άλλη συνάρτηση.
- Δίνουμε τις αρχικές τιμές μέσα σε αγκύλες και δεξιά του πίνακα.
- Χωρίζουμε τις τιμές με κόμμα.



Αρχικές τιμές πίνακα 2/3

Παράδειγμα

```
#include <iostream.h>
#define LIM 6
int x[LIM] = {1, 3, 5, 7, 9, 11}
void main ( )
{
    .....;
}
```



Αρχικές τιμές πίνακα 3/3

- Εάν το πλήθος των αρχικών τιμών είναι μικρότερο των θέσεων του πίνακα, τότε η τιμή των τελευταίων θέσεων του πίνακα θα είναι μηδέν.
- Υπάρχει δυνατότητα να δώσουμε αρχικές τιμές χωρίς να έχουμε ορίσει τη μέγιστη διάσταση του πίνακα.



Βιβλιογραφία

- Jamsa, K. 1999. Εισαγωγή στη... C++. Μετάφραση: Τ. Άλβας. Εκδόσεις Κλειδάριθμος. Αθήνα
- Λάζος, Κ. 2004. C++: Θεωρία και πράξη. 2^η Έκδοση. Θεσσαλονίκη
- Savitch, W. 2013. Πλήρης C++. 4^η έκδοση. Μετάφραση: Σ. Κατσαβούνης. Εκδόσεις Τζιόλα. Αθήνα
- Ανδρεοπούλου, Ζ. 2011. Ηλεκτρονικοί Υπολογιστές. Πηγή στο Διαδίκτυο:
http://www.for.auth.gr/uploads/pages/ΗΛΕΚΤΡΟΝΙΚΟΙ_ΥΠΟΛΟΓΙΣΤΕΣ_2014_theory_and_practise.pdf



Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Ζαχαρούλα Ανδρεοπούλου. «Ηλεκτρονικοί Υπολογιστές. Επαναληπτικές Εντολές στη C++». Έκδοση: 1.0. Θεσσαλονίκη 2015. Διαθέσιμο από τη δικτυακή διεύθυνση: <http://eclass.auth.gr/courses/OCRS351/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά - Παρόμοια Διανομή [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

[1] <http://creativecommons.org/licenses/by-sa/4.0/>





Τέλος ενότητας

Επεξεργασία: <Χριστιάνα Κολιούσκα>
Θεσσαλονίκη, <Χειμερινό εξάμηνο 2014-2015>



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



**ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ**

Σημειώματα

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

