

ΕΠΛ 033: ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΓΙΑ ΜΗΧΑΝΙΚΟΥΣ

Μάριος Belk, Τμήμα Πληροφορικής, Πανεπιστήμιο Κύπρου

Email: belk@cs.ucy.ac.cy





ΠΙΝΑΚΕΣ

Θέματα Διάλεξης

- Δομές Δεδομένων
- Σημασία και Σύνταξη Πινάκων
- Αρχικοποίηση και Διαχείριση Πινάκων
- Πολυδιάστατοι Πίνακες

Γιατί Πίνακες

- Για την ομαδική επεξεργασία δεδομένων που έχουν κοινή σημασία/χρήση
- Παράδειγμα: Ταξινόμηση βαθμολογίας φοιτητών
 - Χρειάζομαι να δηλώσω **X** μεταβλητές τύπου `int`
 - Αντί να δημιουργήσω **X** μεταβλητές `int x1, x2, x3, ... xX;` μπορώ να ζητήσω από τη C να μου κρατήσει **X συνεχόμενες θέσεις** για ακέραιους στη μνήμη
 - Πως;

```
int array[10]; //Δημιουργία πίνακα 10 θέσεων
```

→ Αντί να έχω στη μνήμη

<input type="text"/>	<input type="text"/>	<input type="text"/>	...	<input type="text"/>
x1	x2	x3		x10

→ Θα έχω

array	<input type="text"/>	<input type="text"/>	<input type="text"/>	...	<input type="text"/>
-------	----------------------	----------------------	----------------------	-----	----------------------

Δομές Δεδομένων

4

- Οι πίνακες αποτελούν ένα σημαντικό δομημένο τύπο δεδομένων (structured data type) ή απλά μια δομή δεδομένων (data structure)
- Μία **δομή δεδομένων** είναι ένα σύνολο συγγενικών δεδομένων τα οποία αποθηκεύονται κάτω από το ίδιο όνομα
- Οι πίνακες συνήθως έχουν γρήγορη πρόσβαση και προτιμούνται από άλλες δομές δεδομένων (όταν υπάρχει επιλογή)

Πίνακες

5

- **Πίνακας** είναι μια δομή δεδομένων με τα ακόλουθα χαρακτηριστικά:
 - Καταλαμβάνει **συνεχόμενες θέσεις μνήμης** για διατήρηση: **Συγκεκριμένου αριθμού στοιχείων ΙΔΙΟΥ τύπου**
 - Είναι **στατική δομή**, δηλαδή το μέγεθος τους ορίζεται εκ των προτέρων (πριν την μεταγλώττιση)
- **Ορισμός πίνακα στην C:**

τύπος πίνακα όνομα πίνακα [μέγεθος]
array type array name [positions number]

Ορισμός Πίνακα

6

- Κατά τον **ορισμό ενός πίνακα** θα πρέπει:
 - να καθορίσετε το **όνομα** του
 - Τον **τύπο** του, δηλαδή τον τύπο των στοιχείων του
 - Το **πλήθος των στοιχείων** του

```
arrayType arrayName[ numberOfElements ];
```

→ Παράδειγμα:

```
int c[10]; // ορισμός ενός πίνακα με όνομα c και πλήθος 10 ακεραίων
```

```
float myArray[3284]; /* ορισμός του πίνακα myArray σαν έναν  
πίνακα 3284 πραγματικών αριθμών*/
```

- Ορισμός πολλών πινάκων του ίδιου τύπου
 - Ακολουθούμε τους ίδιους κανόνες, όπως και στην περίπτωση των μεταβλητών.
 - Παράδειγμα:

```
int b[100], x[27];
```

Αρχικοποίηση Πινάκων

7

□ Ταυτόχρονος Ορισμός και Αρχικοποίηση

```
int n[ 5 ] = { 1, 2, 3, 4, 5 };
```

- Όταν οι αρχικές τιμές είναι λιγότερες από ότι οι θέσεις του πίνακα, τότε οι θέσεις που δεν έχουν τιμές είναι οι πιο δεξιές θέσεις. Η εξ' ορισμού τιμή τους είναι το 0

```
int n[ 5 ] = {}; /* ορισμός ενός πίνακα 5 ακεραίων με αρχικές τιμές το 0 για ΌΛΕΣ του τις θέσεις */
```

- Όταν οι αρχικές τιμές είναι περισσότερες από τις θέσεις του πίνακα, τότε προκύπτει συντακτικό λάθος
- Όταν το μέγεθος του πίνακα δεν ορίζεται, τότε αυτό καθορίζεται από το πλήθος των αρχικών τιμών. Για παράδειγμα:

```
int n[ ] = { 1, 2, 3, 4, 5 }; // ορισμός πίνακα 5 ακεραίων με αρχικές τιμές τους αριθμούς από το 1-5
```

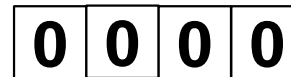
```
char pinakas[]={ 'a', 'b', 'c' }; //ορισμός πίνακα 3 χαρακτήρων με αρχικές τιμές τους χαρακτήρες a, b, c
```


Δήλωση Πίνακα

```
int a[4];
```



```
int b[4]={};
```



```
int c[4]={1,4,3,2};
```



1. Ένας πίνακας με όνομα a με 4 θέσεις ακεραίων
2. Ένας πίνακας με όνομα b με 4 θέσεις ακεραίων
3. Ένας πίνακας με όνομα c με 4 θέσεις ακεραίων

```
int md[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
```

τύπος

όνομα

Μέγεθος/
πλήθος
στοιχείων

md[0]	31
md[1]	28
md[2]	31
...	
md[11]	31

τιμές

Πρόσβαση Πίνακα

9

- Σύνταξη: όνομα [αριθμοδείκτης]
- **Ο αριθμοδείκτης πίνακα (subscript ή index) πρέπει να είναι τύπου int**
 - $x[0] \Rightarrow$ τιμή πρώτου στοιχείου του πίνακα x
 - $x[i] \Rightarrow$ τιμή i -οστού συν ένα (i th +1) στοιχείου του πίνακα x
- Τα στοιχεία σε ένα πίνακα με n στοιχεία αναφέρονται ως στοιχείο 0, στοιχείο 1, στοιχείο 2, ..., και στοιχείο ($n-1$). Πχ στο md το στοιχείο 0 έχει τιμή 31, το στοιχείο 1 τιμή 28 κτλ
- **Ευθύνη προγραμματιστή να επαληθεύσει ότι η τιμή αριθμοδείκτη πίνακα είναι στο πεδίο $[0, \text{πλήθος_στοιχείων} - 1]$**

Επεξεργασία ανά Στοιχείο

- Μπορούμε να θεωρήσουμε **καθένα από τα στοιχεία ενός πίνακα σαν απλή μεταβλητή**. Δηλαδή, μπορούμε να του αναθέσουμε τιμή, να διαβάσουμε την τιμή αυτή και να αλλάξουμε την τιμή του. Για παράδειγμα:
- `int x[]={1,2,3,5,7,11,13,17,19,23,29,31};`
 - `printf ("%d", x[0]);`
 - `x[3] = 8258;`
 - `sum = x[0] + x[1];`
 - `x[3] += 1;`
 - `x[2] = x[0] + x[1];`
 - `x[2] = x[3] + 1;`

Αναφορά σε Στοιχεία του Πίνακα

11

- $x[i] = 0;$
- $x[i] = x[j];$
- $x[j+k*4] = x[u] + 3;$
- $x[x[i]] = p;$
- $\text{diff} = x[y] - x[z];$
- $\text{total} = \text{total} + x[i++];$
- $\text{float } f = 5.0;$
 $(x[i] == x[(\text{int})f])$

Αρχικοποίηση Πίνακα 1

12

```
#define MAX 100
```

```
/*αρχικοποίηση στοιχείων του πίνακα x σε 0*/
```

```
int x[MAX], i;
```

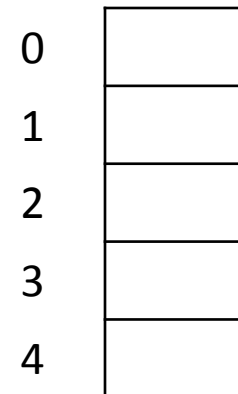
```
for(i=0;i<MAX;++i)
```

```
    x[i]=0;
```

Αρχικοποίηση Πίνακα 2

13

```
#define SIZE 5  
int square[SIZE], i;  
/*αρχικοποίηση:στοιχείο i σε i^2*/  
for (i = 0; i < SIZE; ++i)  
    square[i] = i * i;
```



Πίνακες και βρόχοι επανάληψης

14

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

```
int main(){
```

```
    // dilwsi kai arxikopoiisi pinaka
```

```
    double temp[5] = {12.4, 19.6, 23.8, 28.9, 30.2};
```

```
    /* me xrisi vroxou epanalipsis ektyponoume stin othoni
```

```
    * ta stoixeia tou pinaka */
```

```
    for (int i=0; i<5; i++) {
```

```
        printf("%.2f\n", temp[i]);
```

```
    }
```

```
}
```

Λίστα αρχικοποίησης όλων των στοιχείων του πίνακα



Ενδεικτική Επίλυση

```
C:\Users\admin\Dropbox\EPL033-2015\20...
12.40
19.60
23.80
28.90
30.20
Press any key to continue . . .
```

Πίνακες και βρόχοι επανάληψης

15

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

```
int main(){
```

```
// dilwsi pinaka 10 thesewn
```

```
int ar[10];
```

```
/* me xrisi vroxou epanalipsis thetoyme times
```

```
* apo to 0 mexri to 9 sta stoixeia tou pinaka */
```

```
for (int i=0; i<10; i++) {
```

```
    ar[i]=i;
```

```
}
```

```
/* me xrisi vroxou epanalipsis ektyponoyme tis times
```

```
* twv stoixeiwn tou pinaka */
```

```
for (int i=0; i<10; i++) {
```

```
    printf("%d %d\n", i, ar[i]);
```

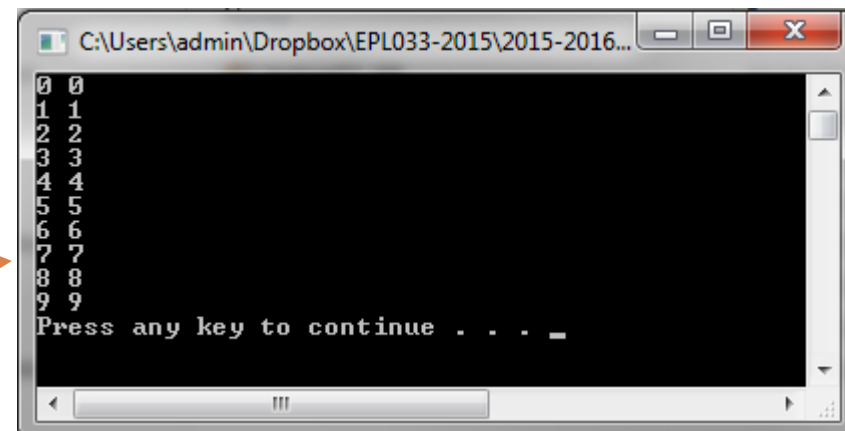
```
}
```

```
}
```

Αρχικοποίηση των τιμών του πίνακα μέσα από ένα for βρόχο

Εκτύπωση των θέσεων και των τιμών των στοιχείων του πίνακα

Ενδεικτική εκτέλεση



```
C:\Users\admin\Dropbox\EPL033-2015\2015-2016...
0 0
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
Press any key to continue . . . _
```


Πίνακες και βρόχοι επανάληψης

16

```
#include <stdio.h>
#define SIZE 10 // megethos pinaka
```

Η οδηγία #define χρησιμοποιείται για να ορίσουμε την τιμή της ΣΤΑΘΕΡΑΣ Length σε 10 κατά το στάδιο της προεπεξεργασίας πριν την μεταγλώττιση.

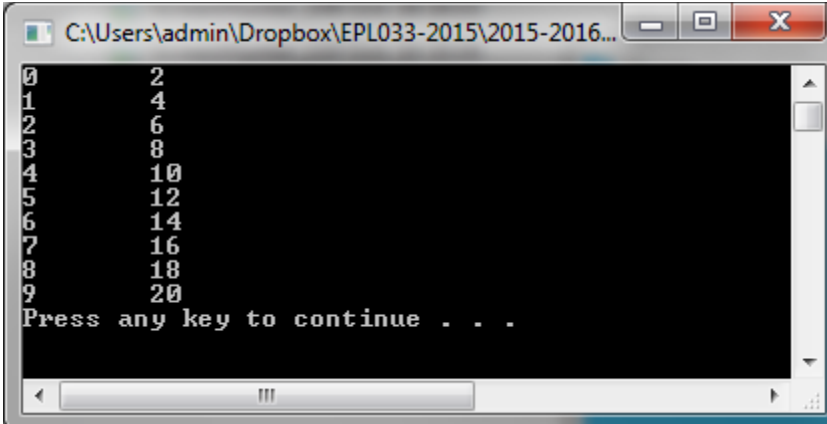
```
int main(){
    // dilwsi pinaka SIZE thesewn
    int ar[SIZE];

    /* me xrisi vroxou epanalipsis thetoyme times
    * 2+2*i sta stoixeia tou pinaka */
    for (int i=0; i<SIZE; i++) {
        ar[i]=2+2*i;
    }

    /* me xrisi vroxou epanalipsis ektyponoyme tis times
    * twn stoixeiwn tou pinaka */
    for (int i=0; i<SIZE; i++) {
        printf("%d\t%d\n", i, ar[i]);
    }
}
```

Η τιμή του Length αντικαθιστάται με 10 από τον μεταγλωττιστή. Συνεπώς, ο πίνακας ar έχει 10 στοιχεία

Ενδεικτική Επίλυση



```
C:\Users\admin\Dropbox\EPL033-2015\2015-2016...
0      2
1      4
2      6
3      8
4     10
5     12
6     14
7     16
8     18
9     20
Press any key to continue . . .
```

Πίνακες

17

```
#include <stdio.h>
#define SIZE 10 // megethos pinaka

int main(){
    // dilwsi pinaka SIZE thesewn
    int ar[SIZE];
    int ar2[SIZE];

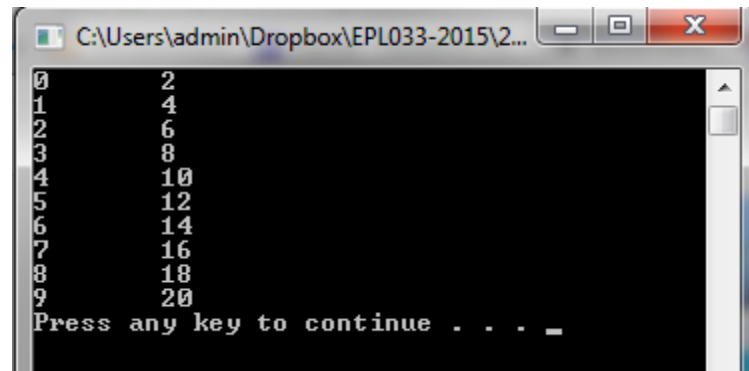
    /* me xrisi vroxou epanalipsis thetoyme times
    * 2+2*i sta stoixeia tou pinaka */
    for (int i=0; i<SIZE; i++) {
        ar[i]=2+2*i;
    }

    /* me xrisi vroxou epanalipsis antigrafw
    * tis times toy prwtoy pinaka ston deytero
    * kai ektypwnw tis times toy deyteroy */
    for (int i=0; i<SIZE; i++) {
        ar2[i]=ar[i];
        printf("%d\t%d\n", i, ar2[i]);
    }
}
```

Σε περίπτωση που θέλουμε να αντιγράψουμε ένα πίνακα σε ένα άλλο θα πρέπει να μεταφέρουμε τα στοιχεία **ένα προς ένα!**

Οι πίνακες ΔΕΝ αντιγράφονται σαν ένα στοιχείο αφού το όνομα του πίνακα είναι ουσιαστικά απλά μια σταθερά διεύθυνσης

Για να το πετύχουμε αυτό χρησιμοποιούμε έναν for βρόχο που θα διαβάζει ένα στοιχείο σε κάθε επανάληψη και θα το μεταφέρει στο νέο πίνακα



```
C:\Users\admin\Dropbox\EPL033-2015\2...
0      2
1      4
2      6
3      8
4     10
5     12
6     14
7     16
8     18
9     20
Press any key to continue . . . _
```

Μέσος Όρος Στοιχείων Πίνακα

18

```
#include <stdio.h>
#define SIZE 10 // megethos pinaka

int main(){
    // dilwsi pinaka SIZE theswn
    int ar[SIZE];
    int sum=0;

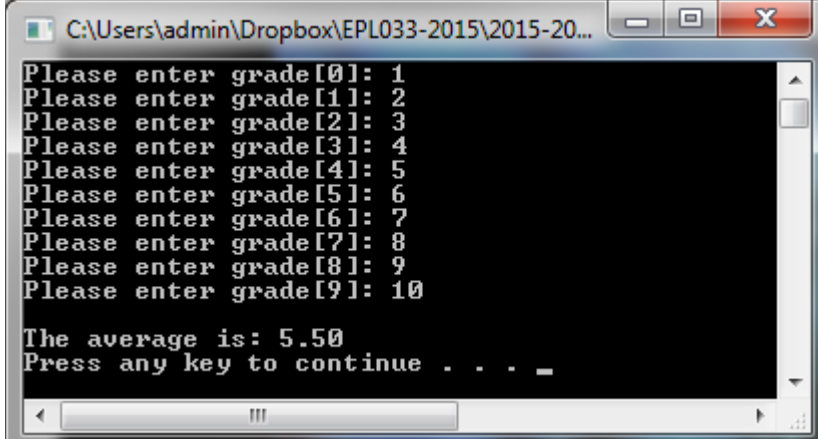
    // eisagwgi timwn apo ton xristi
    for (int i=0; i<SIZE; i++) {
        printf("Please enter grade [%d]\n: ", i);
        scanf("%d", &ar[i]);
    }

    // ypologismos athroismatos
    for (int i=0; i<SIZE; i++) {
        sum = sum + ar[i];
    }

    // ypologismos mesou orou
    float mesos_oros = (float) sum/SIZE;

    // ektypwsi apotelesmatos
    printf("The average is: %.2f\n", mesos_oros);

    return 0;
}
```



```
C:\Users\admin\Dropbox\EPL033-2015\2015-20...
Please enter grade[0]: 1
Please enter grade[1]: 2
Please enter grade[2]: 3
Please enter grade[3]: 4
Please enter grade[4]: 5
Please enter grade[5]: 6
Please enter grade[6]: 7
Please enter grade[7]: 8
Please enter grade[8]: 9
Please enter grade[9]: 10

The average is: 5.50
Press any key to continue . . . _
```

Πίνακες – σύγκριση στοιχείων

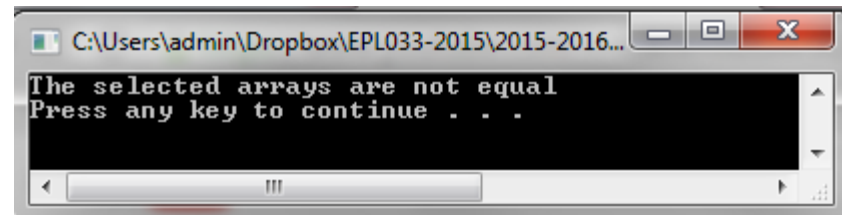
19

```
#include <stdio.h>
#define SIZE 5 // megethos pinaka

int main(){
    // dilwsi kai arxikopoiisi pinakwn SIZE thesewn
    int ar[]={2,3,5,7,9};
    int ar2[]={2,3,5,9,7};
    // metavliti pou tin allazoume an yparxoun anisa stoixeia
    int einai_isa = 1;

    // gia kathe thesi
    for (int i=0; i<SIZE; i++) {
        // an DEN einai isa ta antistoixa stoixeia
        // metevale tin metavliti einai_isa
        if (ar[i] != ar2[i]) {
            einai_isa = 0;
        }
    }

    // ektypwsi apotelesmatos
    if ( einai_isa == 1 )
        printf("The selected arrays are equal");
    else
        printf("The selected arrays are not equal");
}
```



```
C:\Users\admin\Dropbox\EPL033-2015\2015-2016...
The selected arrays are not equal
Press any key to continue . . .
```

Παράλληλοι Πίνακες

20

```
#define STUDENT_NUM 55  
int student_id[STUDENT_NUM];  
float student_grade[STUDENT_NUM];
```

- `student_id[i]` περιέχει αρ. ταυτότητας και `student_grade[i]` τον βαθμό του φοιτητή με ταυτότητα `student_id[i]`
- Οι πίνακες με ίδιο αριθμό στοιχείων χρησιμοποιούνται (συνήθως) για συγγενικές πληροφορίες

Παράδειγμα

21

	student_id
0	12345
1	37349
2	9995
...	
54	20001

	student_grade
0	2.12
1	6.14
2	4.56
...	
54	7.8

Κώδικας

22

```
#define STUDENT_NUM 55
int main()
{
    int student_id[STUDENT_NUM];
    float student_grade[STUDENT_NUM];
    int j;
    ... /* dinontai times stous pinakes */
    for(j=0;j<size;++j)
        printf("Student with id: %d, grade: %f\n",
            student_id[j],student_grade[j]);
    return 0;
}
```

Παράλληλοι Πίνακες

23

- Εναλλακτικά ένας πίνακας με εγγραφές (structures)

	student_id	student_grade
0	12345	2.12
1	37349	6.14
2	9995	4.56
...		
54	20001	7.8

- Θα τις μελετήσουμε σε μελλοντική Ενότητα

Πολυδιάστατοι Πίνακες (Multidimensional arrays)

- πίνακας με δύο ή περισσότερες διαστάσεις
 - ▣ π.χ. `float student_grades[NUM_STUDENTS][NUM_COURSES];`

		course			
		0	1	2	3
student	0				
	1				
	2				
	3				
	4				
	5				

Δυσδιάστατος πίνακας
NUM_STUDENTS γραμμές
NUM_COURSES στήλες

Σύνταξη και Σημασιολογία

- Σύνταξη δήλωσης:
τύπος ονομα[dim1][dim2]..[dimn];
- Σημασία: πίνακας με n διαστάσεις και $(dim1 * dim2 * .. * dimn)$ στοιχεία
- Όνομα περιέχει την διεύθυνση του 1ου στοιχείου
 - π.χ. **student_grades** περιέχει $NUM_STUDENTS * NUM_COURSES$ στοιχεία
 - 1ο στοιχείο: **student_grades[0][0]**
 - τελευταίο: **student_grades[NUM_STUDENTS-1] [NUM_COURSES-1]**

Σημασιολογία

26

- Ένας άλλος τρόπος να δούμε έναν δυσδιάστατο πίνακα:
 - ▣ ένας πίνακας του οποίου κάθε στοιχείο είναι πίνακας
- Με αυτό τον τρόπο εξηγούνται πιο εύκολα διάφορα χαρακτηριστικά των πολυδιάστατων πινάκων:
 - ▣ Ο τρόπος αρχικοποίησης στην δήλωση
 - ▣ Ο τρόπος παράθεσης στην μνήμη
 - ▣ κ.α.

Επεξεργασία Πολυδιάστατων Πινάκων

- Η επεξεργασία πολυδιάστατου πίνακα απαιτεί φωλιασμένους βρόχους (το επίπεδο φωλιάσματος είναι ίσο με τον αριθμό διαστάσεων του πίνακα)

- Αρχικοποίηση

- **Στον ορισμό**

- `int count[4][3]={ {4,5,6}, {3,56,87}, {456,412,846}, {0,0,0} };`

- πίνακας 4 (γραμμές) x 3(στήλες)

- **Με φωλιασμένο βρόχο**

```
for(i=0;i<NUM_STUDENTS;++i)
    for(j=0;j<NUM_COURSES;++j)
        student_grades[i][j]=0;
```

ομαδοποίηση σε γραμμές

Πολυδιάστατοι Πίνακες και Συναρτήσεις

- Ορισμός Συνάρτησης με multi-dim παράμετρο:
 - ▣ `void display(int table[][NUM_COURSES], ...);`
 - ▣ μόνο το μέγεθος της 1ης διάστασης δε χρειάζεται (γιατί;)
- Κλήση συνάρτησης με multi-dim παράμετρο
 - ▣ `display(student_grades, ...);`
 - ▣ Στην κλήση περνάμε σαν παράμετρο μόνο το όνομα του πίνακα και εάν χρειαστεί και τις διαστάσεις σαν ξεχωριστές μεταβλητές.
 - Ισχύει το ίδιο και στην κλήση συναρτήσεων που έχουν ως μεταβλητή εισόδου μονοδιάστατο πίνακα.

Αποθήκευση Πολυδιάστατων Πινάκων

- Τα δεδομένα πολυδιάστατου πίνακα αποθηκεύονται στη μνήμη με *προτεραιότητα σειράς*: 1η γραμμή, 2^η γραμμή, 3η γραμμή, κτλ.

```
int count[4][3]=
```

```
{ {4,5,6},
```

```
{3,56,87},
```

```
{456,412,846},
```

```
{0,0,0} };
```

0,0	4
0,1	5
0,2	6
1,0	3
1,1	56
1,2	87

....

Κλασσικά Παραδείγματα

- Πρόσθεση πινάκων $n \times m$
 - ▣ Πρόσθεση *αντίστοιχων* στοιχείων
- Πολλαπλασιασμός πινάκων $n \times k, k \times m$
 - ▣ Πράξεις μεταξύ γραμμών του πρώτου και στήλες του δεύτερου (όπως ακριβώς ισχύει και στα μαθηματικά)

Παράδειγμα

- Γράψετε ένα πρόγραμμα:
 - ▣ Να διαβάζει από την είσοδο βαθμούς 80 φοιτητών. Για κάθε φοιτητή διαβάζονται 5 βαθμοί. Τα δεδομένα είναι καταχωρημένα ως ακολούθως:
90 60 70 89 45 (βαθμοί 1ου φοιτητή) 40 56 78 99 100 (βαθμοί 2ου φοιτητή) κτλ.
 - ▣ Να υπολογίζει και να τυπώνει
 - (α) βαθμούς κάθε φοιτητή,
 - (β) τον μέσο όρο για κάθε φοιτητή, και
 - (γ) τον μέσο όρο για κάθε μάθημα

Τι πρέπει να γίνει;

- Διαβαστούν τα δεδομένα (αποθηκευτούν)
 - ▣ **void read_data(float grades_table[80][5]);**
- Υπολογισμός μέσων όρων
 - ▣ **void compute_averages(float grades_table[80][5],
float average_per_student[80],
float average_per_course[5]);**
- Εκτύπωση αποτελεσμάτων
 - ▣ **void display_results(float grades_table[80][5],
float average_per_student[80],
float average_per_course[5]);**

Τι πρέπει να γίνει; (συν.)

σταθερές

```
#define NUM_STUDENTS 80  
#define NUM_COURSES 5
```

πίνακες στη main():

```
float grade_table[NUM_STUDENTS][NUM_COURSES];  
float average_per_student[NUM_STUDENTS];  
float average_per_course[NUM_COURSES];
```

Τι πρέπει να γίνει; (συν.)

```
void read_data(float grade_table[][NUM_COURSES])
{
    int i,j;
    for(i=0;i<NUM_STUDENTS;++i)
        for(j=0;j<NUM_COURSES;++j)
            scanf("%f",&grade_table[i][j]);
}
```

Τι πρέπει να γίνει; (συν.)

```
void read_data(float grade_table[][NUM_COURSES])
{
    int i,j;
    FILE *fdat;
    fdat = fopen("input.dat", "r");
    for(i=0;i<NUM_STUDENTS;++i)
        for(j=0;j<NUM_COURSES;++j)
            fscanf(fdat, "%f", &grade_table[i][j]);
    fclose(fdat);
}
```

```
void compute_averages(const float grade_table[][NUM_COURSES],
float average_per_student[], float average_per_course[]) {
int i, j;

for(i=0; i<NUM_STUDENTS; ++i)
    average_per_student[i]=0; //Αρχικοποίηση πίνακα average_per_student
for(i=0; i<NUM_COURSES; ++i)
    average_per_course[i]=0; //Αρχικοποίηση πίνακα average_per_course
for(i=0; i<NUM_STUDENTS; ++i)
    for(j=0; j<NUM_COURSES; ++j){
        average_per_student[i]+=grade_table[i][j];
        average_per_course[j]+= grade_table[i][j];
    }
for(i=0; i<NUM_STUDENTS; ++i)
    average_per_student[i]/=NUM_COURSES;
for(i=0; i<NUM_COURSES; ++i)
    average_per_course[i]/=NUM_STUDENTS;
}
```

Τι πρέπει να γίνει; (συν.)

```
void display_results(float grade_table[][NUM_COURSES],
float average_per_student[],
float average_per_course[]){
int i,j;
for(i=0;i<NUM_STUDENTS;++i){
    for(j=0;j<NUM_COURSES;++j)
        printf("%5.2f ", grade_table[i][j]); //student's grades
    printf("%5.2f\n",average_per_student[i]; //student's average
}
for(j=0;j<NUM_COURSES;++j)
    printf("%5.2f ",average_per_course[j]);
printf("\n");
}
```

Πιο γενικό πρόβλημα

- Τα δυο πρώτα δεδομένα ορίζουν τον αριθμό φοιτητών και αριθμό μαθημάτων (που δεν πρέπει να ξεπερνά το 80 και το 5 αντίστοιχα). Τι αλλαγές συνεπάγονται από το πιο πάνω;

Τι πρέπει να γίνει;

```
void read_data(float grade_table[][NUM_COURSES],
int students, int courses){
int i,j;
scanf("%d %d",&students, &courses);
if (students>NUM_STUDENTS || courses>NUM_COURSES){
    printf("error: students or courses > than allowed maximum");
    exit(0);
}
for(i=0;i<students;++i)
    for(j=0;j<courses;++j)
        scanf("%f",&grade_table[i][j]);
}
```


Τι πρέπει να γίνει; (συν.)

```
void compute_averages(float grade_table[][NUM_COURSES],
float average_per_student[], float average_per_course[], int students, int courses) {
    inti,j;
    for(i=0;i<students;++i)
        average_per_student[i]=0;
    for(i=0;i<courses;++i)
        average_per_course[i]=0;
    for(i=0;i<students;++i)
        for(j=0;j<courses;++j){
            average_per_student[i]+=grade_table[i][j];
            average_per_course[j]+= grade_table[i][j];
        }
    for(i=0;i<students;++i)
        average_per_student[i]/=courses;
    for(i=0;i<courses;++i)
        average_per_course[i]/=student;
}
```

Τι πρέπει να γίνει; (συν.)

```
void display_results(float grade_table[][NUM_COURSES],
float average_per_student[], float average_per_course[],
int students, int courses){
int i,j;
for(i=0;i<students;++i){
    for(j=0;j<courses;++j)
        printf("%5.2f ",grade_table[i][j]);
    printf("%5.2f\n",average_per_student[i];
}
for(j=0;j<courses;++j)
    printf("%5.2f ",average_per_course[j]);
printf("\n");
}
```

Περίληψη

- Δομές Δεδομένων
- Πίνακες
 - ▣ Σημασία
 - ▣ Σύνταξη
 - ▣ Αποθήκευση και πρόσβαση πινάκων
- Διαχείριση Στοιχείων Πινάκων
- Πολυδιάστατοι Πίνακες