ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά μαθήματα ΠΠ

ΑΡΧΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Κεφάλαιο 11

Επιμέλεια:
Βασίλης Παλιουράς , Αναπληρωτής Καθηγητής
Ευάγγελος Δερματάς , Αναπληρωτής Καθηγητής
Σταύρος Νούσιας , Βοηθός Ερευνητή
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών & Τεχνολογίας Υπολογιστών

Άδειες Χρήσης

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

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου των διδασκόντων καθηγητών.
- Το έργο «Ανοικτά Ακαδήμαϊκά Μαθήματα στο Πανεπιστήμιο Πατρών» έχει χρηματοδοτηθεί μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκη Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικους πόρους.

Ανάπτυξη

Το παρόν εκπαιδευτικό υλικό αναπτύχθηκε στο τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών του Πανεπιστημίου Πατρών

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Πίνακας αλφαριθμητικών

```
#define N 3
#include <stdio.h>

main ( ) {
 int i ;
 char text[N][11] ={"dokimi", "test", "paradeigma"};

 printf("text requires %d bytes\n", sizeof text ) ;

 for (i=0 ; i < N ; printf ("%s ", text[i++]));

}
```

Ο πίνακας 2-D λειτουργεί ως πίνακας 1-D με στοιχεία πίνακες 1-D.

text[0]	‘d’	‘o’	‘k’	‘i’	‘m’	‘i’	0				
text[1]	‘t’	‘e’	‘s’	‘t’	0						
text[2]	‘p’	‘a’	‘r’	‘a’	‘d’	‘e’	‘i’	‘g’	‘m’	‘a’	0

Πίνακας αλφαριθμητικών

```
#define N 3
#include <stdio.h>

main ( ) {
 int i ;
 char text1[N][11] ={"dokimi", "test", "paradeigma"};
 char *text2[N] = {"dokimi", "test", "paradeigma"};

 printf("text1 requires %d bytes\n", sizeof text1 ) ;
 printf("text2 requires %d bytes\n", sizeof text2 );

for (i=0 ; i < N ; printf ("%s ", text2[i++]))
    ;

}
```

Πίνακας 1-D με στοιχεία δείκτες σε χαρακτήρα

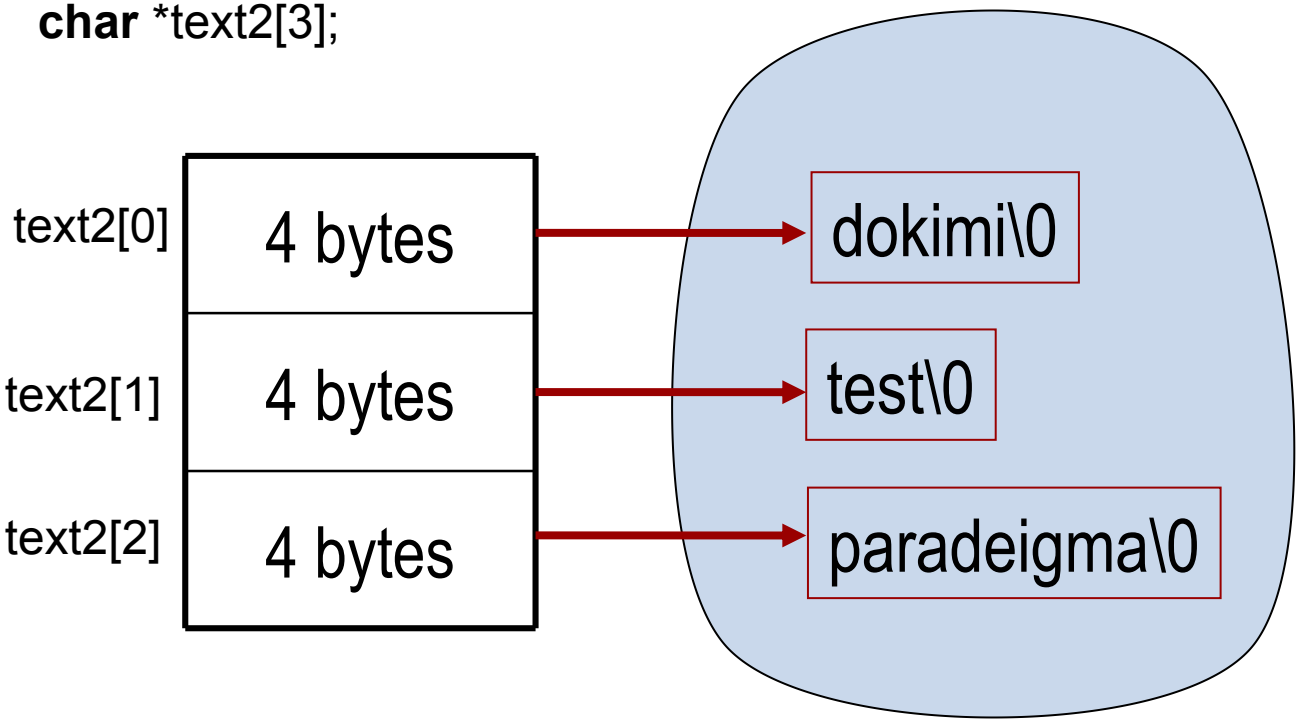
Προσοχή:
Δεν περιλαμβάνει τις αλφαριθμητικές σταθερές!

```
$ ./a
text1 requires 33 bytes
text2 requires 12 bytes
dokimi test paradeigma
Paliuras@MOB ~/nlect9
$
```

Οργάνωση μνήμης και πίνακες δεικτών σε χαρακτήρες

Οι αλφαριθμητικές σταθερές της αρχικοποίησης αποθηκεύονται σε άλλη περιοχή μνήμης.

char *text2[3];



text2: περιλαμβάνει τρεις διευθύνσεις, όχι τα ίδια τα αλφαριθμητικά.

Δήλωση και αρχικοποίηση

```
main ( ) {
    char name[10] = "katerina";

    printf ("%s", name);

    scanf("%s", name);

    printf ("%s", name);

}
```

Δήλωση και ανάθεση τιμής σε αλφαριθμητικό

```
char *strcpy(char *, const char*);
```

```
main ( ) {
   char name[10];

   name = "katerina";

   printf ("%s", name);

   scanf("%s", name);

   printf ("%s", name);

}
```

Λάθος (στη C)

```
#include <string.h>
main ( ) {
   char name[10];

   strcpy(name,"katerina");
   printf ("%s", name);

   scanf("%s", name);
   printf ("%s", name);

}
```

Σωστός τρόπος:

Τι γίνεται με δείκτες;

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

main ( ) {
char *name =
"katerina";

printf ("%s", name);


}
```

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

main ( ) {
char *name ;

name =
"katerina";

printf ("%s",
name);
}
```

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

main ( ) {
char *name ;

name = "katerina";

printf ("%s", name);

*name = 'K';

printf ("%s", name);

}
```

σφάλμα χρόνου εκτέλεσης (run-time error):
segmentation fault

Λύση: χρήση διαθέσιμης περιοχής μνήμης (πχ με calloc ())

Διεύθυνση ονόματος πίνακα

- `char word[5];`
- `word` είναι η διεύθυνση του πρώτου στοιχείου
  `&word[0] => char *`
- &word
  - έχει αριθμητικά την ίδια τιμή με το word
  - Είναι όμως τύπου => `char (*)[5]`

Παράδειγμα

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

int main(int argc, char *argv[]) {

   int array[5] = { 0, 1, 2, 3, 4} ;

   printf("Have the same value...\n");
   printf("array : %X\n", array);
   printf("&array: %X\n", &array);

   printf("...but not the same type:\n");
   printf("next element  (array+1): %X\n", array + 1);
   printf("next element (&array+1): %X\n", &array + 1 );

   return 0;
}
```

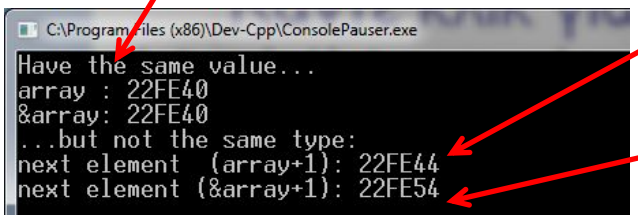
array, &array
Έχουν ***ίδια*** τιμή,
Αλλά ***διαφορετικό*** τύπο

- **int** *
- **int** (*) [5]

Ίδια τιμή

Μια θέση ακεραίου μετά

20 θέσεις ακεραίου μετά
Δηλ. (1 πίνακας array)

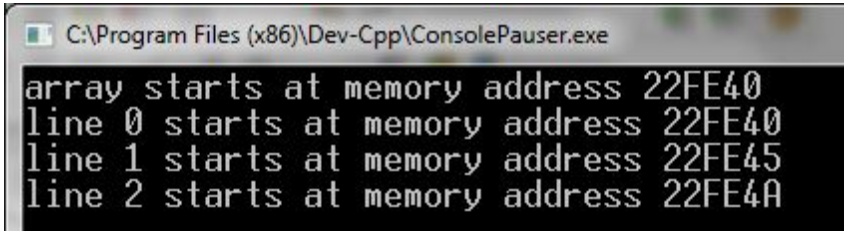


Πίνακας χαρακτήρων

```
#include <stdio.h>

int main(int argc, char
*argv[]) {
 int i;
 char manywords[3][5];

 printf("array starts at memory
address %X\n", manywords);

 for (i = 0; i < 3; i++)
   printf("line %d starts at
memory address %X\n",
                i,
manywords[i]);

 return 0;
}
```

Πίνακας δύο διαστάσεων

- **char** manywords[3][5];
- Ένας πίνακας 3 στοιχείων,
  - καθένα από τα οποία είναι πίνακας 5 στοιχείων,
    - καθένα από τα οποία είναι **char**.



Διεύθυνση αρχής της i-οστής γραμμής

Πίνακες πολλών διαστάσεων ως παράμετροι σε συναρτήσεις

```
char a[3][2];
```

[0][0]	[0][1]
[1][0]	[1][1]
[2][0]	[2][1]

```
char b[2][3];
```

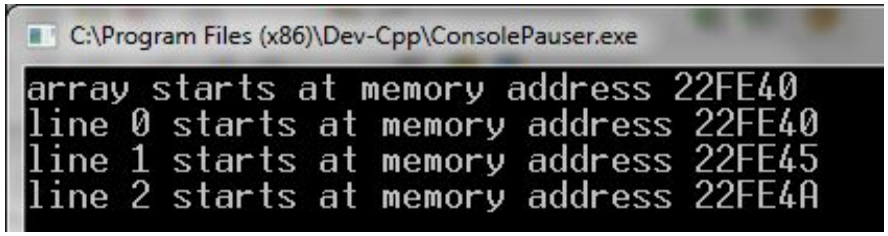
[0][0]	[0][1]	[0][2]
[1][0]	[1][1]	[1][2]

- Οι συναρτήσεις χειρίζονται τους πίνακες κατ' αναφορά
- Η θέση ενός στοιχείου στη μνήμη εξαρτάται από τη γεωμετρία του πίνακα
  - Το a[1][0] είναι το 3ο στοιχείο, το b[1][0] είναι το 4ο στοιχείο
- Μια συνάρτηση πρέπει να ξέρει τη γεωμετρία ενός πίνακα παραμέτρου
  - Μπορούμε να παραλείψουμε μόνο την πρώτη διάσταση (το πλήθος γραμμών) σε μια δήλωση
    - Σε πίνακες N διαστάσεων, μπορούμε να παραλείψουμε μόνο μία διάσταση και να δηλώσουμε μήκη για τις υπόλοιπες N – 1
- Ευέλικτος κώδικας χρησιμοποιώντας ως παράμετρο δείκτη στο πρώτο στοιχείο του πίνακα

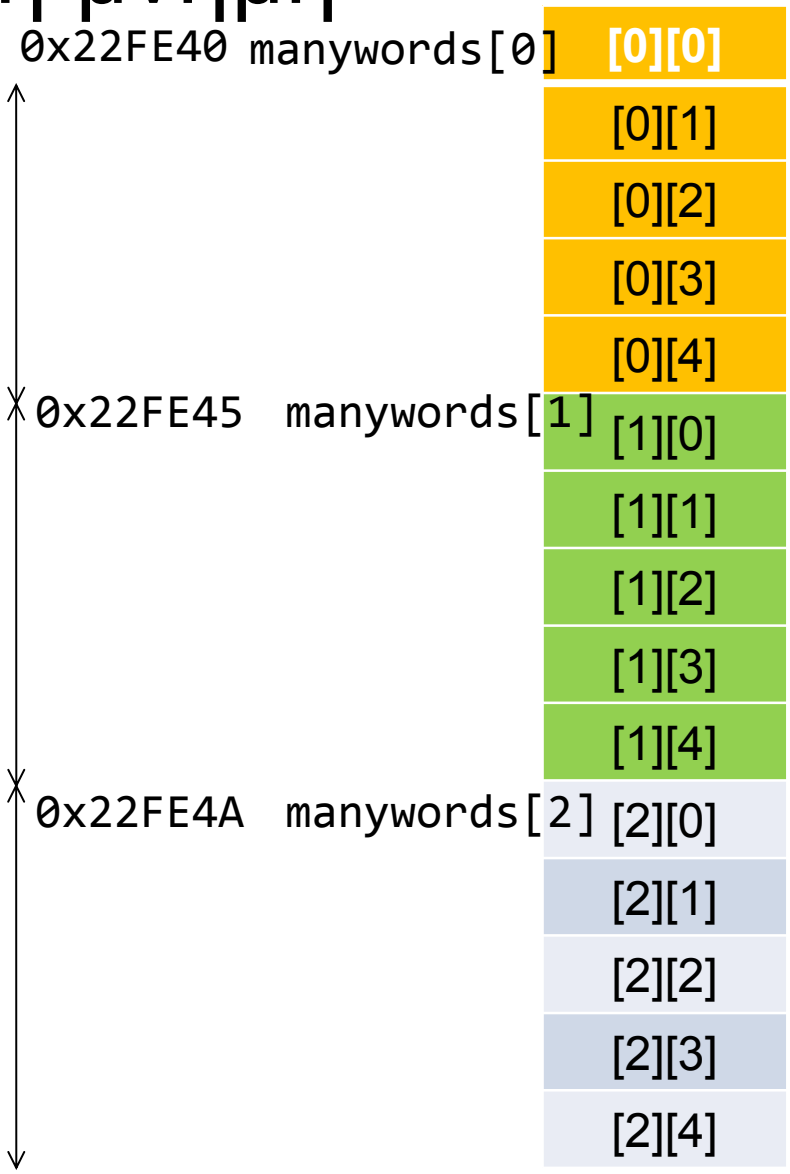
Αποθήκευση στη μνήμη

```
char manywords[3][5];
```

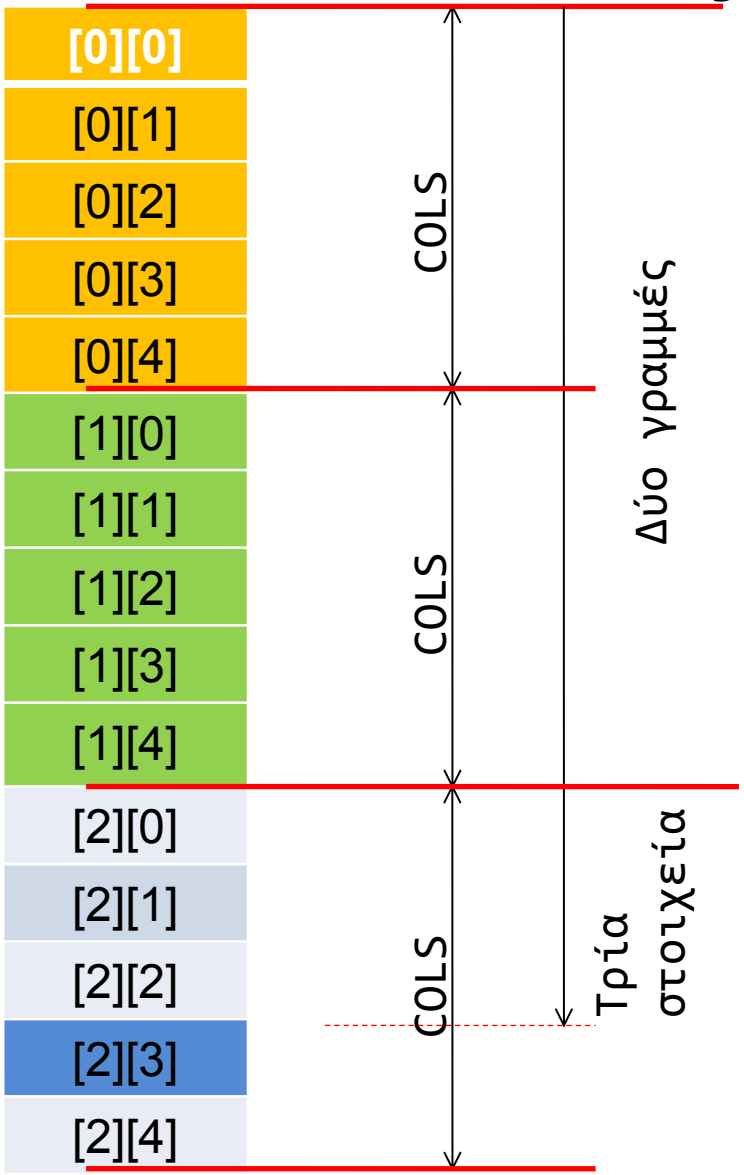
[0][0]	[0][1]	[0][2]	[0][3]	[0][4]
[1][0]	[1][1]	[1][2]	[1][3]	[1][4]
[2][0]	[2][1]	[2][2]	[2][3]	[2][4]



Το μέγεθος σε bytes ανά στοιχείο καθορίζεται από τον τύπο του στοιχείου
Εδώ **char**, άρα 1 byte ανά στοιχείο.



Πίνακας δύο διαστάσεων




```
#define ROWS 3
#define COLS 5
char manywords[ROWS][COLS];
```

[0][0]	[0][1]	[0][2]	[0][3]	[0][4]
[1][0]	[1][1]	[1][2]	[1][3]	[1][4]
[2][0]	[2][1]	[2][2]	[2][3]	[2][4]

Κάθε γραμμή έχει COLS στοιχεία

Το στοιχείο `manywords[2][3]` βρίσκεται δύο γραμμές (άρα 2 * 5 στοιχεία) συν τρία στοιχεία της τρέχουσας γραμμής από την αρχή του πίνακα

Το στοιχείο `manywords[i][j]` βρίσκεται στη θέση

```
(char *)manywords+ i*COLS+j
```

Πίνακες πολλών διαστάσεων

- `int numbers[2][3][4];`
- Ένας πίνακας 2 στοιχείων,
  - καθένα από τα οποία είναι πίνακας 3 στοιχείων,
    - καθένα από τα οποία είναι πίνακας 4 στοιχείων
      - καθένα από τα οποία είναι **`int`**

Πίνακες πολλών διαστάσεων

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

void readarray(int *, int, int, int);
void printarray(int *, int, int, int);
void print3D(int [2][3][4]);


int main(int argc, char *argv[]) {

        int array3D[2][3][4];

        readarray( (int *)array3D, 2, 3, 4);
        printarray((int *)array3D, 2, 3, 4);
        print3D(array3D);

        return 0;
}
```

Εδώ η γεωμετρία του πίνακα είναι παράμετρος

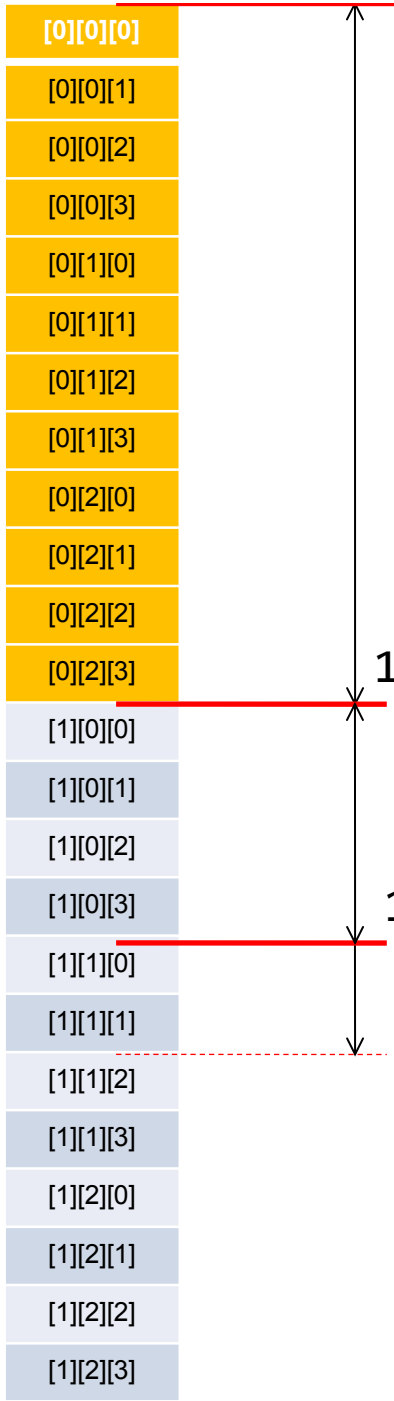
Χειρίζεται πίνακες τύπου **int** [2][3][4] μόνο

Πίνακας σταθερής γεωμετρίας ως παράμετρος

```
void print3D(int data[2][3][4]) {
        int i, j, k;

        for (i = 0; i<2; i++) {
                for (j=0; j<3; j++) {
                        for (k=0; k<4; k++)
                                printf("%2d ", data[i][j][k]);
                        printf("\n");
                                }
                printf("\n");
                }

}
```

Χειρίζεται πίνακες τύπου **int** [2][3][4] μόνο

Παράδειγμα




```
int array3D[2][3][4];
            h  w  d

(int *) array3D + i*w*d + j * d + k
```

Το στοιχείο array3D[1][1][2] βρίσκεται στη θέση
(int *) array3D + 1*3*4+ 1*4 + 2

Πίνακας σταθερού πλήθους διαστάσεων, όχι προκαθορισμένης γεωμετρίας, ως παράμετρος σε συνάρτηση

```
void readarray(int *data, int h, int w, int d) {
int i, j, k ,count = 0  ;
        for (i = 0; i<h; i++)
                for (j=0; j<w; j++)
                        for (k=0; k<d; k++)
                                *(data + i*w*d + j * d + k) = count ++;
}
void printarray(int *data, int h, int w, int d) {
int i, j, k ;
        for (i = 0; i<h; i++) {
                for (j=0; j<w; j++) {
                        for (k=0; k<d; k++)
                                printf("%2d ", *(data + i*w*d + j*d + k));
                        printf("\n");
                                }
                printf("\n");
                }
}
```

Αναφερόμαστε στο στοιχείο [i][j][k] ενός πίνακα [h][w][d]

Παράδειγμα

```
void myread(int *data, int h, int w, int d) {
  int i=0, count = 0;
  for (; i< h*w*d; *(data + (i++)) = count ++ );
}
void myread(int *data, int h, int w, int d) {
  int i=0;
  for (; i< h*w*d; *(data + i) = i++ );
}
```

```
void myread(int *data, int h, int w, int d) {
  int i=0;
  for (; i< h*w*d; *(data ++) = i++ );
}
```

```
void myread(int *data, int h, int w, int d) {
  int i=0;
  for (; (*(data ++) = i++ )< h*w*d;);
}
```

Σε ***μερικές*** περιπτώσεις μπορούμε να κάνουμε αντίστοιχα πράγματα
Με λιγότερο κώδικα, αξιοποιώντας τον τρόπο αποθήκευσης στη μνήμη.

Ευέλικτος κώδικας

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

void readarray(int *, int, int, int);
void printarray(int *, int, int, int);
void print3D(int [2][3][4]);


int main(int argc, char *argv[]) {

        int array3D[2][3][4];
        int another3D[2][2][2];

        readarray( (int *)array3D, 2, 3, 4);
        printarray((int *)array3D, 2, 3, 4);
        print3D(array3D);

        readarray( (int *)another3D, 2, 2, 2);
        printarray((int *)another3D, 2, 2, 2);
        return 0;
}
```

Ίδια συνάρτηση, η γεωμετρία ως του πίνακα ως παράμετρος.

Πίνακες πολλών διαστάσεων

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

int main(int argc, char *argv[])
{
  int array2d[][4] = {{1,2,3,4}, {5,6,7,8}, {9,10,11,12}};
  int i, j;

  printf("Size of array in bytes: %d\n", sizeof array2d);

  printf("Size of an element: %d\n", sizeof array2d[0]);

  for (i=0; i< 3; i++)
      for (j=0; j< 4; j++)
          printf("%d %d %d\n", i, j, array2d[i][j]);


  return 0;
}
```

48 bytes

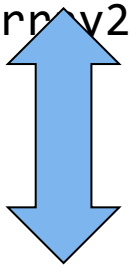
16 bytes

Πίνακες πολλών διαστάσεων

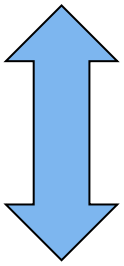
```
for (i=0; i< 3; i++)
      for (j=0; j< 4; j++)
          printf("%d %d %d\n", i, j, *( array2d[i] + j));
```

```
for (i=0; i< 3; i++)
      for (j=0; j< 4; j++)
          printf("%d %d %d\n", i, j, *( *(array2d + i) + j));
```

```
for (i=0; i< 3; i++)
      for (j=0; j< 4; j++)
          printf("%d %d %d\n", i, j, *((int *) array2d + i * 4 +
   j));
```



Το i μετράει γραμμές



4 ακέραιοι ανά γραμμή

Αναλυτική αλλαγή τύπου σε (int *)

⇒ Γίνεται διεύθυνση ακεραίου

Το C99 επιτρέπει Variable Length Arrays (VLAs)

```
void test(int rows, int cols, int x[rows][cols]) {
/* … */
}
```

Παράδειγμα

```
#include <stdio.h>
#include <stdlib.h>
//
// C99 comment style
// Demonstration of C99 VLAs as parameters in functions
//
void readarray(int rows, int cols, int x[rows][cols]);
void printarray(int rows, int cols, int x[rows][cols]);

int main(int argc, char *argv[]) {
              int a[5][5];
              readarray(5,  5, a);
              printarray(3, 5, a);
              return 0;
}

void readarray(int rows, int cols, int x[rows][cols]) {
    int i, j;

    for (i = 0; i< rows; i++)
              for (j = 0; j < cols; j++)
                                        x[i][j] = -i*cols - j ;
}

void printarray(int rows, int cols, int x[rows][cols]) {
    int i, j;

    for (i = 0; i< rows; i++) {
              for (j = 0; j < cols; j++)
                  printf("%4d", x[i][j]) ;
              printf("\n");
              }
}
```

Δεν υποστηρίζονται
Από όλους τους
Compilers!!!

Το mvsc δεν είναι
C99 compiler

Εδώ γράφουμε C90
Εκτός αν ζητείται αναλυτικά

Σημείωμα αναφοράς

- Copyright Πανεπιστήμιο Πατρών, Παλιουράς Βασίλειος , Δερματάς Ευάγγελος «Αρχές Προγραμματισμού ». Έκδοση: 1.0. Πάτρα 2015
- Διαθέσιμο από τη δικτυακιακή διέυθυνση https://eclass.upatras.gr/modules/