



**Τμήμα Μηχανικών Η/Υ & Πληροφορικής
Πανεπιστήμιο Πατρών**

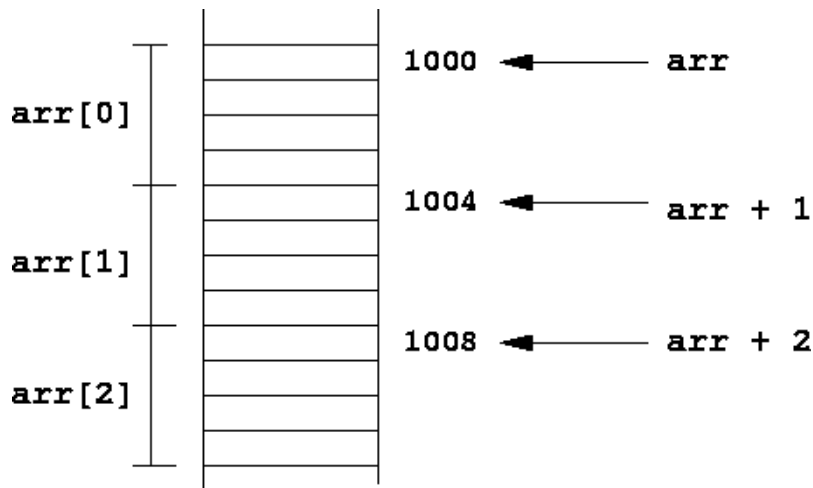
Εισαγωγή στον Προγραμματισμό

Η γλώσσα προγραμματισμού C

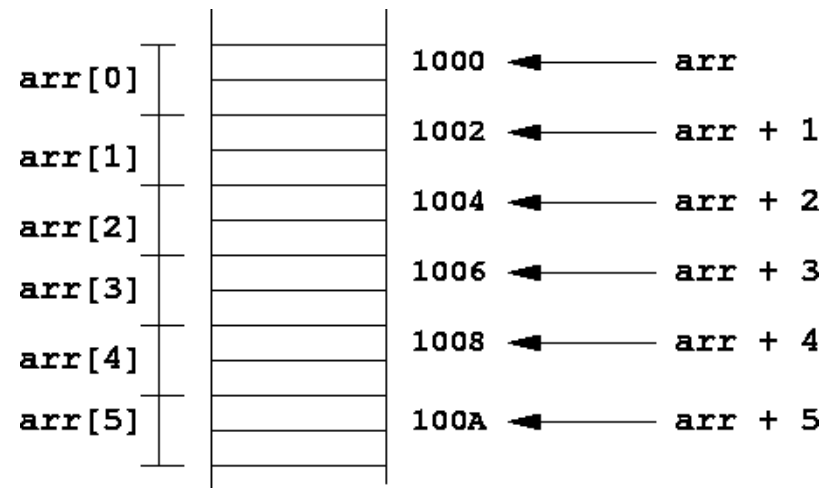
Pointer Arithmetic (Αριθμητική Δεικτών)

Το κατά πόσο αυξάνεται/μειώνεται ένας pointer σε πράξεις με ακεραίους, εξαρτάται από τον τύπο του.

int arr[10];
sizeof(int) == 4



short arr[10];
sizeof(short) == 2



Δείκτες – Πίνακες

- Το όνομα ενός πίνακα είναι μία «σταθερά τύπου δείκτη» με τιμή τη διεύθυνση του πρώτου στοιχείου του πίνακα.
- Λόγω αυτού του γεγονότος μπορούμε να ορίσουμε ένα δείκτη στον πίνακα και να χειριστούμε τον πίνακα μέσω αυτού του δείκτη.
- Δεν είναι εφικτή η ανάθεση τιμής και η αριθμητική με δείκτες

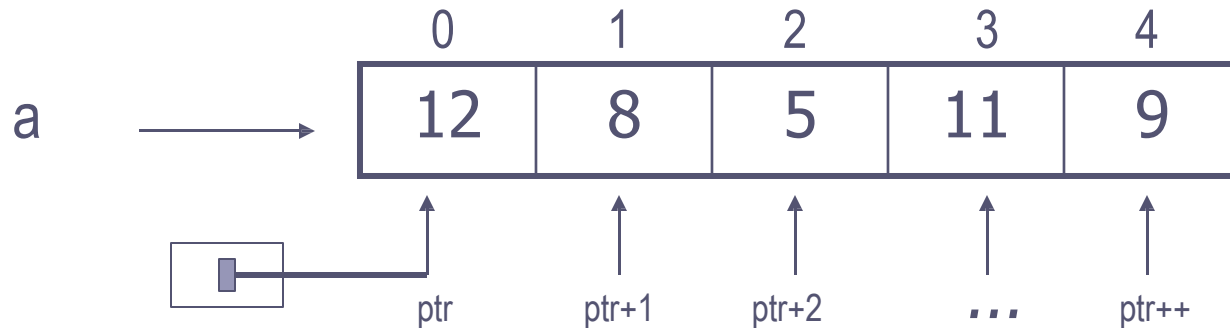
Δείκτες – Πίνακες

Έστω

```
int a[5] = { 12, 8, 5, 11, 9};
```

```
int *ptr;
```

```
ptr = a; ή ptr = &a[0];
```

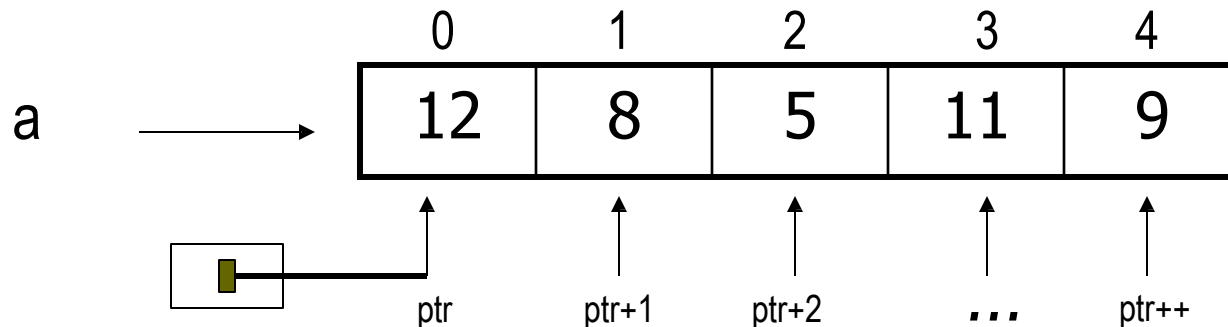


Προσοχή: `a++`; `a--` (το όνομα του πίνακα δεν είναι δείκτης (δηλ. μεταβλητή δείκτη))

Δείκτες – Πίνακες

- Μπορούμε να δώσουμε σε έναν pointer την διεύθυνση ενός array, και στη συνέχεια να τον χρησιμοποιήσουμε ως array

```
int a[] = {12, 8, 5, 11, 9}, *ptr;  
ptr=a;  
printf("%d\n", ptr[1]);    // 8  
ptr += 2;  
printf("%d\n", ptr-a);    // 2  
printf("%d\n", ptr[1]);    // 11  
ptr--;  
printf("%d\n", ptr[1]);    // 5  
a++;    // Compile error, arr is const!
```



Δυναμική Διαχείριση Μνήμης

- ***Calloc, Malloc,***
για δέσμευση μνήμης
- ***Realloc***
για αλλαγή μεγέθους δεσμευμένης μνήμης
- ***Free***
για απελευθέρωση μνήμης

Malloc()

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

```
void *malloc(size_t size);
```

Η συνάρτηση malloc δεσμεύει χώρο μεγέθους τουλάχιστον size bytes.

Επιστρέφει ένα δείκτη στη δεσμευθείσα μνήμη ή NULL, αν δεν υπάρχει διαθέσιμη μνήμη. Τα bytes του χώρου μνήμης δεν παίρνουν αρχική τιμή

```
x=(int *)malloc(n*sizeof(int)); // int *x;
```

```
x=(int **)malloc(n*sizeof(int *));
```

```
for (i=0; i<n; i++) *(x+i)=(int *)malloc(m*sizeof(int));
```

```
&x[i][j] ---- *(x+i)+j
```

```
x[i][j] ---- *(* (x+i)+j)
```

Calloc()

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

```
void *calloc(size_t n, size_t size);
```

Η συνάρτηση calloc δεσμεύει χώρο για ένα πίνακα n στοιχείων, μεγέθους size το καθένα. Διασφαλίζει την αρχικοποίησή της με 0.

Επιστρέφει ένα δείκτη στη δεσμευθείσα μνήμη ή NULL, αν δεν υπάρχει διαθέσιμη μνήμη.

```
p=(int *)calloc(n, sizeof(int));
```


Realloc()

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

```
void *realloc(void *buffer, size_t size);
```

Η συνάρτηση `realloc` μεταβάλλει το μέγεθος ενός τμήματος μνήμης που είχε προηγουμένα δεσμευτεί. Το νέο μέγεθος ορίζεται από τη `size`. Διασφαλίζει τα υπάρχοντα περιεχόμενα στη μνήμη. Επιστρέφει ένα δείκτη στο νέο τμήμα μνήμης που μπορεί να είναι ίδιος με τον `buffer` ή διαφορετικός αν το τμήμα μετακινηθεί. Αν ο `buffer` είναι `NULL`, η `realloc` λειτουργεί σαν τη `malloc`.

```
p=(int *) realloc(p, n*sizeof(int));
```

Free ()

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

```
void free(void * buffer);
```

Η συνάρτηση `free` αποδεσμεύει τη μνήμη η οποία σηματοδοτείται από το όρισμα `buffer` και η οποία είχε προηγουμένα δεσμευτεί με κλήση μίας εκ των *calloc*, *malloc*, *realloc*.