

# Εισαγωγή στην Πληροφορική & στον Προγραμματισμό

---

Αρχές Προγραμματισμού Η/Υ (με τη γλώσσα C) – **Εργαστηριακή online συνάντηση**

Διάλεξη #7

Πέμπτη, 12 Μαΐου 2022

Παναγιώτης Παύλου

[c-programming-22@allos.gr](mailto:c-programming-22@allos.gr)

# Ανακεφαλαίωση

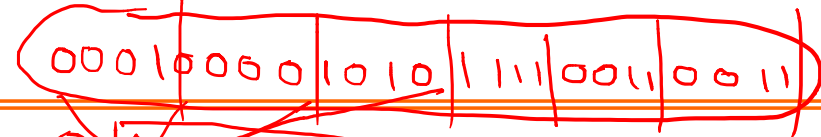
---

Με μία ματιά να θυμηθούμε τη θεωρία

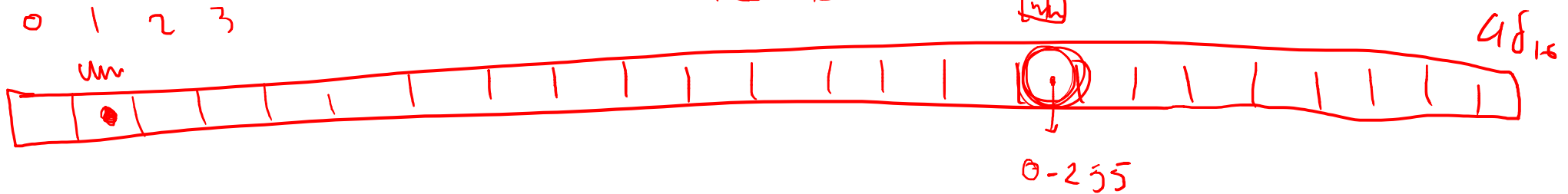
# Πρόχειρο

'A'	1	double 8
int	4	1e-3
short	2	

0x38 38

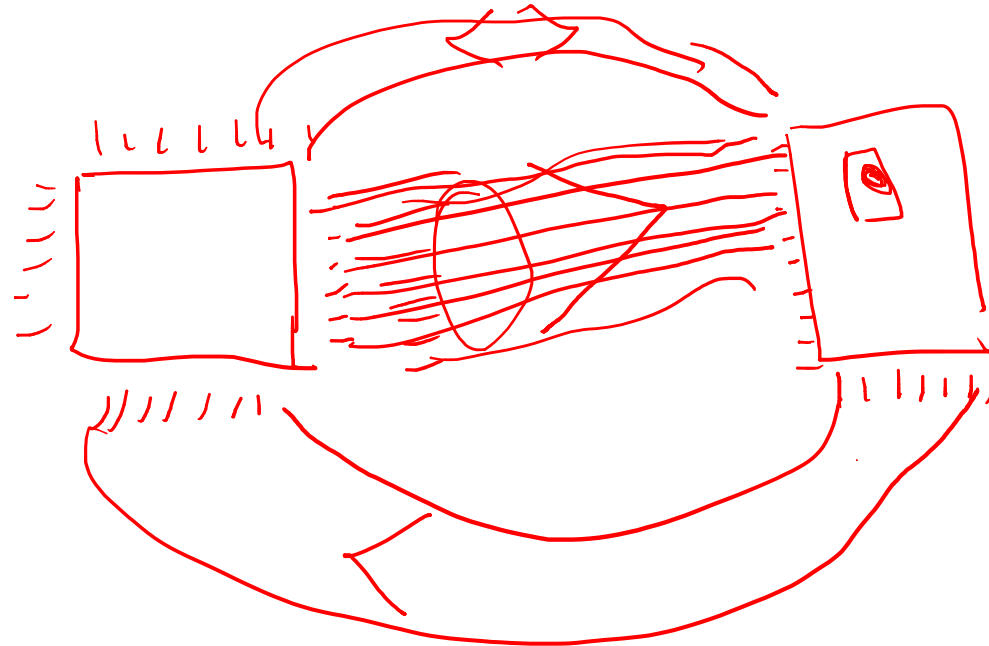


4GB ~ 4 GiB



10110011

address bus



# ΔΕΪΚΤΕΣ

ampersant

int 32bit = 4B = 4 θ.Μ.

int x = 8;  
int x;  
x = 8;

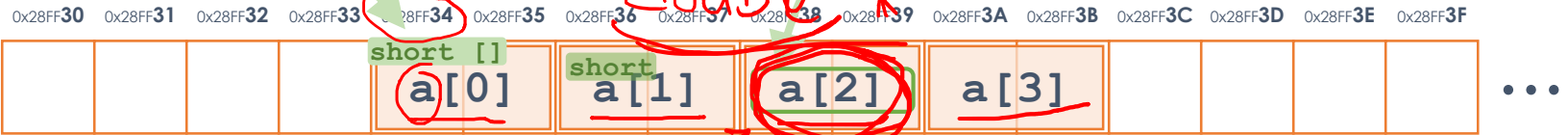
int \*Ap;

```
int *Ap = &a;
```



- a: Τιμή a  
- Ap: \*Ap / &a

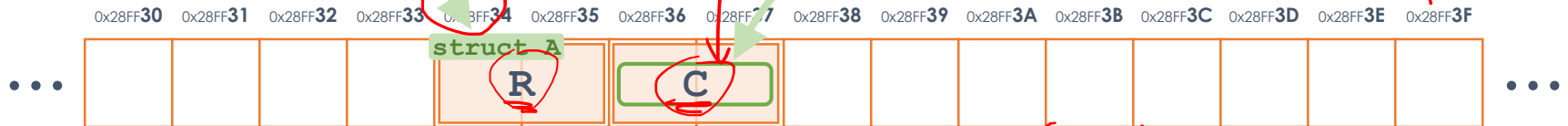
```
short *Ap = a;
```



a[2] Ap[2]

&a[2]

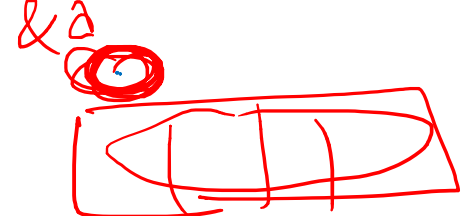
```
struct A *Ap = &a;
```



a.C  
Ap -> C

(\*Ap).C

# Πρόχειρο



int a;

f(&a);

void f(int \*Ap) {

~~a~~ \*Ap = 100;

}

a ~~Ap~~  
 div(5, 2, a, b)

← div(int A, int P,  
int \*PML, int \*YPOL)

T <sub>εξω</sub>	T <sub>γιαω</sub>	
	Τίμη	Pointer
a	a	&a
Ap	*Ap	Ap

# Δείκτες 2

Μεταβλητές

Πίνακες

typedef struct \_CAT {  
int ~;  
double ~;  
} Cat; Δομές

SMTypε

Για αυτά τα δεδομένα

```
varType var;
```

```
varType array[100];
```

```
struct someType aStruct;
```

Μια κατάλληλη μεταβλητή pointer

```
varType *pointer;
```

```
varType *arrPtr;
```

```
struct someType *structPtr;
```

Εκχώρηση του pointer των δεδομένων στη μεταβλητή

```
pointer = &var;
```

```
arrPtr = array;
```

```
structPtr = &aStruct;
```

Χρήση της τιμής μέσω του pointer

```
*pointer
```

```
arrPtr[2]
```

```
structPtr->fieldName
```

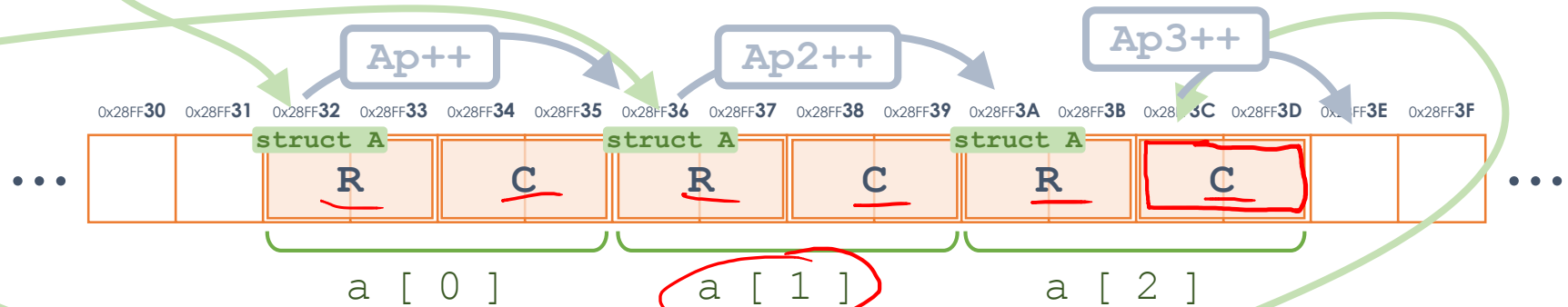
(\*pointer).field

(\*stPtr).field

# Αριθμητική

```
struct A {  
    short R, C;  
};
```

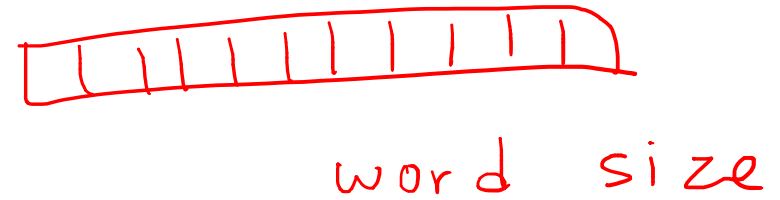
```
struct A a[3]  
struct A *Ap = a;
```



```
struct A *Ap2 = &a[1];
```

```
short *Ap3 = &a[2].C;
```

# Αριθμητική 2



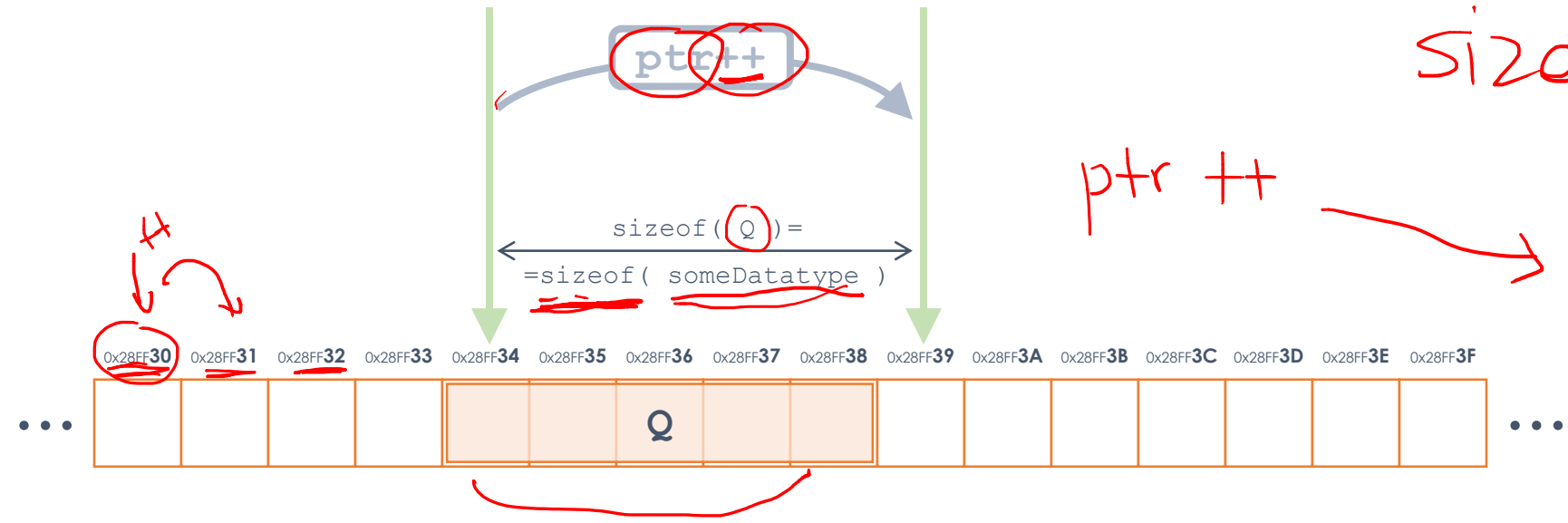
--

```
someDatatype Q;  
someDatatype *ptr = &Q;
```

ptr = 0x1000

sizeof(Q) = 4

ptr ++  
→ 0x1004



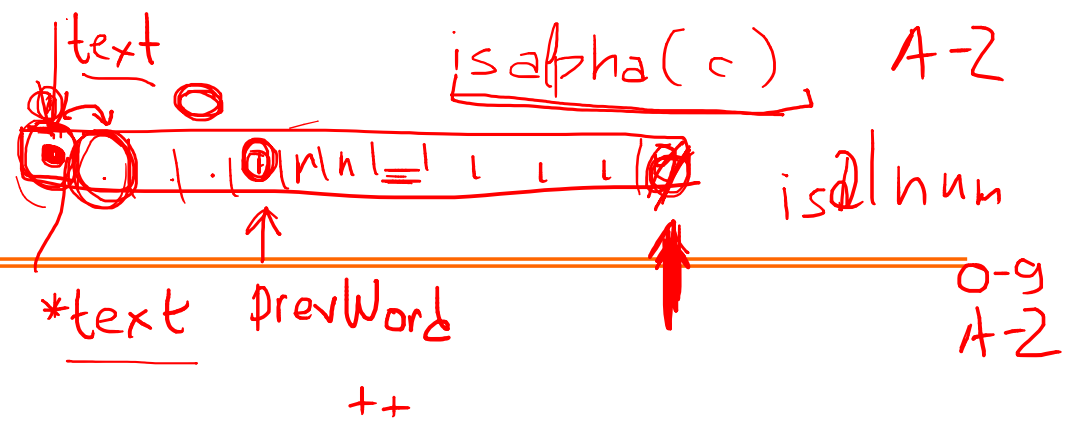


# Εφαρμογές

---

Ας εφαρμόσουμε επιτέλους τη θεωρία

# Επόμενη Λέξη



Το ζητούμενο είναι να γράψουμε μια συνάρτηση, την:

```
char *nextWord(char *text)
```

η οποία να επιστρέφει τον pointer στο σημείο του κειμένου text που ξεκινάει η πρώτη λέξη. Εάν δεν βρεθεί κάποια λέξη τότε πρέπει να επιστρέφεται η τιμή NULL.

Κατόπιν τροποποιήστε την παραπάνω συνάρτηση ώστε όταν αντί ως κείμενο text δίνεται η τιμή NULL να συνεχίζει την αναζήτηση στο προηγούμενο κείμενο που είχε δοθεί, για να βρει την επόμενη λέξη.

Τέλος τροποποιήστε την παραπάνω συνάρτηση η οποία να παίρνει και το όρισμα L όπως φαίνεται και μέσω αυτού να επιστρέφει το μήκος της λέξης που βρέθηκε. Εάν ως L δοθεί το NULL, τότε δεν χρειάζεται να υπολογίζει το μήκος της λέξης. Εάν όμως δοθεί και δεν βρεθεί λέξη, τότε θα πρέπει το μήκος να παίρνει την τιμή 0.

```
char *nextWord(char *text, int *L)
```

Η παραπάνω διαδικασία της σταδιακής συγγραφής της συνάρτησης απεικονίζει την ανάπτυξη της επίλυσης των προβλημάτων. Επιλύοντάς τα ένα προς ένα μπορούμε να ελέγχουμε την ορθότητα του κώδικά μας

# Ερωτήσεις?

---

- Διαβάστε τις σημειώσεις, διαβάστε τις διαφάνειες και δείτε τα videos **πριν** ρωτήσετε
- **Συμβουλευτείτε** τη σελίδα ερωταποκρίσεων του μαθήματος  
<https://qna.c-programming.allos.gr>
- **Στείλτε** τις ερωτήσεις σας πριν και μετά το μάθημα στο  
[c-programming-22@allos.gr](mailto:c-programming-22@allos.gr)
- Εάν έχετε **πρόβλημα** με κάποιο κώδικα στείλτε μαζί τον κώδικα και τα μηνύματα λάθους από το CLI ως κείμενα με copy/paste. Εάν θεωρείτε ότι επιπλέον βοηθά και ένα στιγμιότυπο οθόνης, είναι καλοδεχούμενο.
- Επαναλαμβάνουμε : Μην στείλετε ποτέ κώδικα ως εικόνα μας είναι παντελώς άχρηστος!

