

# Εισαγωγή στην Πληροφορική & στον Προγραμματισμό

---

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

Διάλεξη #7

11 & 12 Μαΐου 2023

Παναγιώτης Παύλου

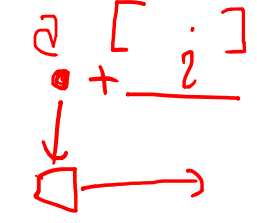
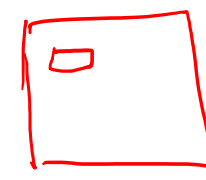
[c-programming-23@allos.gr](mailto:c-programming-23@allos.gr)

# Ανακεφαλαίωση

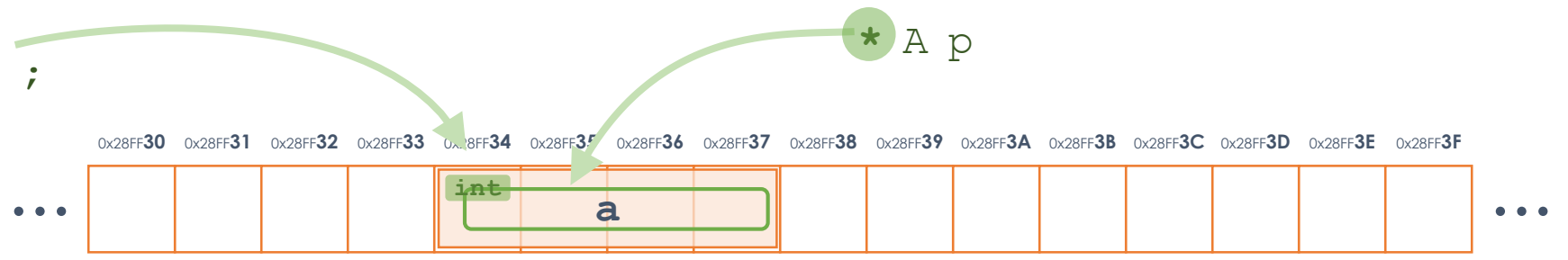
---

Με μία ματιά να θυμηθούμε τη θεωρία

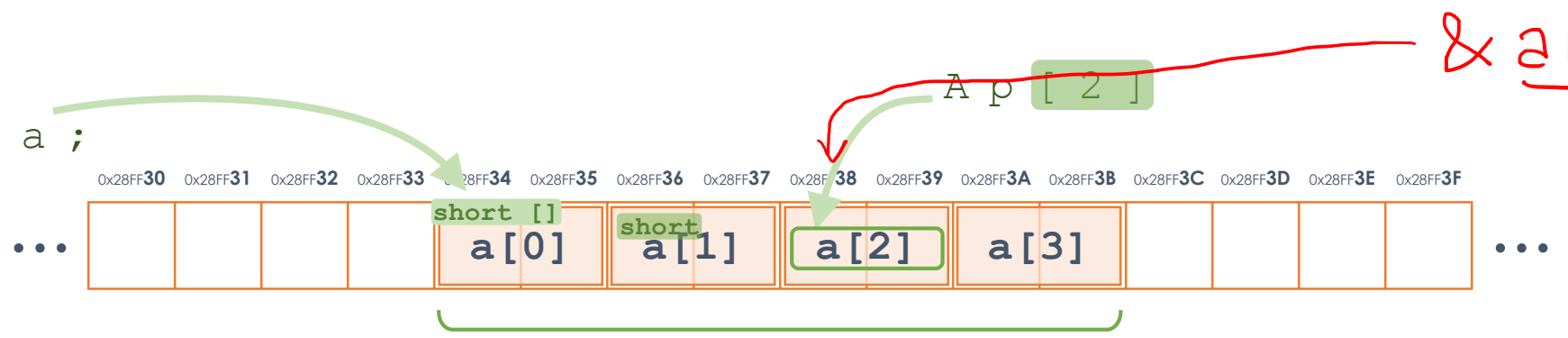
# ΔΕΪΚΤΕΣ



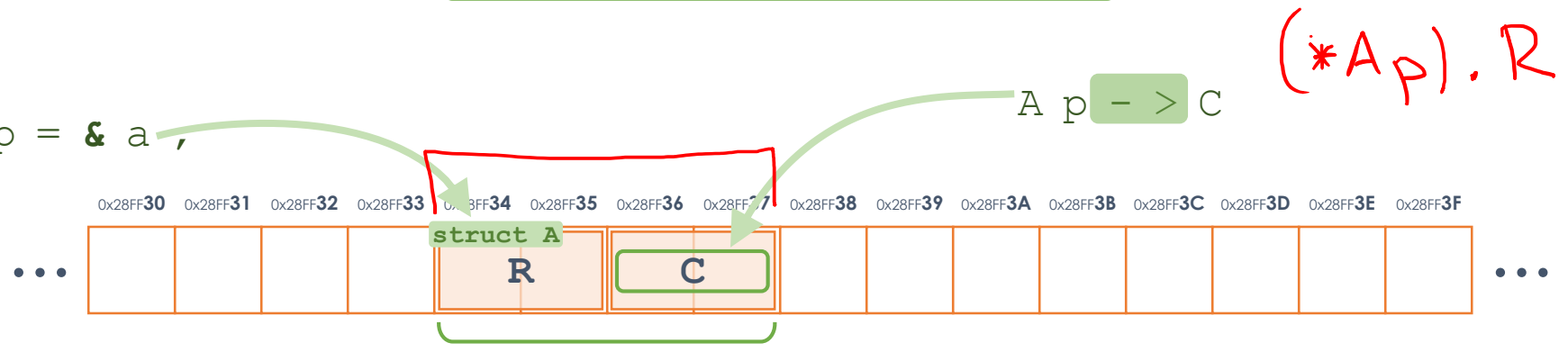
```
int *Ap = &a;
```



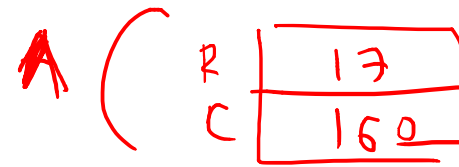
```
short *Ap = a;
```



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



# Δείκτες 2



$$A_p = \&A$$

$$A.C$$
$$A_p \rightarrow C$$

Μεταβλητές

Πίνακες

Δομές

Γι 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

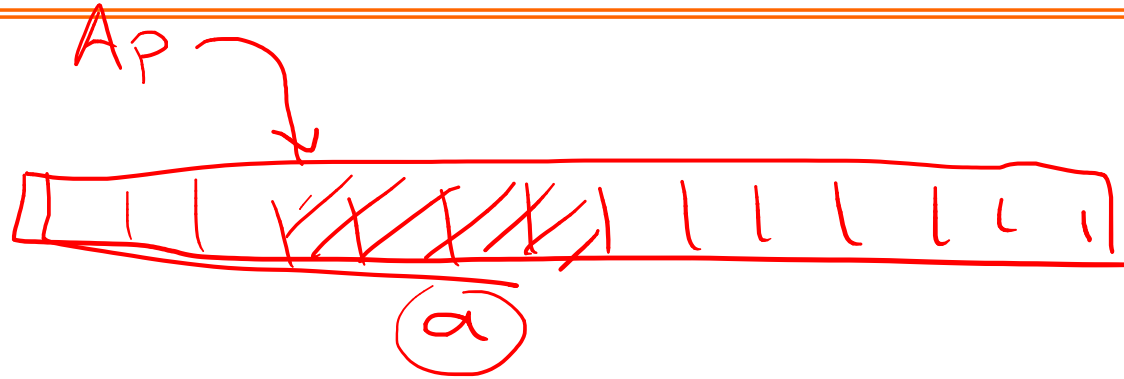
```
*pointer
```

```
arrptr[2]
```

```
structPtr -> fieldName
```

# Πρόχειρο

$A_p ++$



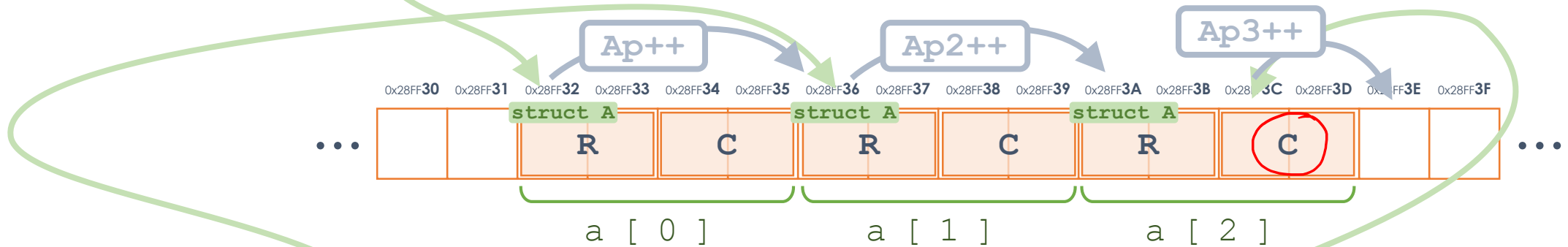
$\text{sizeof } a$

$\underbrace{*(A_p + i)}$

$\underbrace{A_p[i]}$

# Αριθμητική

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

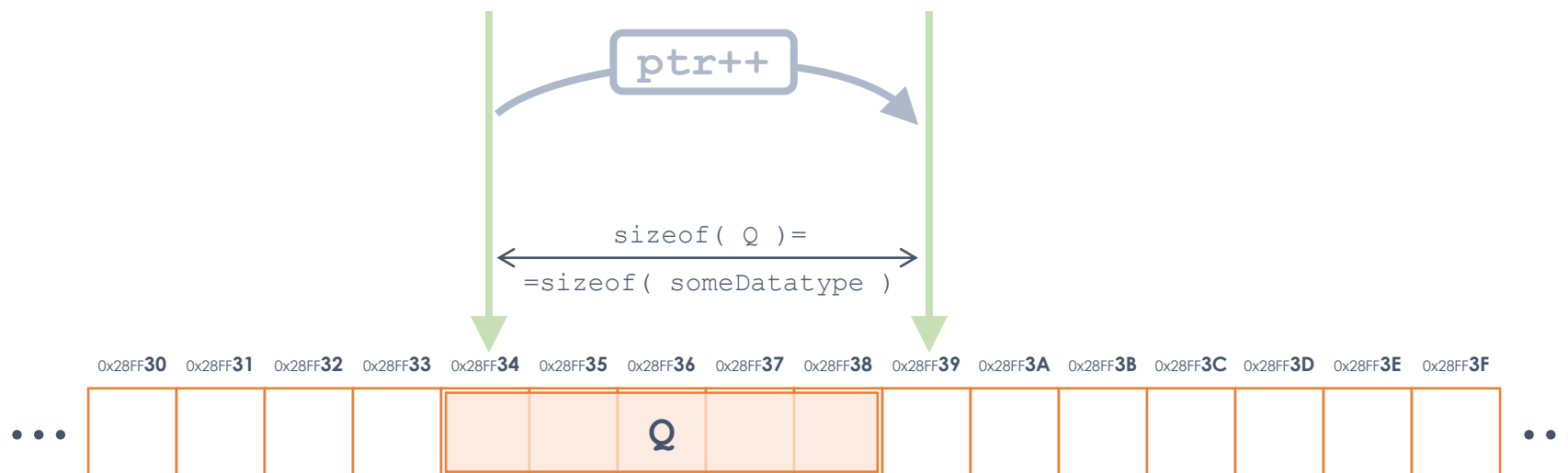


```
struct A *Ap2 = &a[1]
```

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

# Αριθμητική 2

```
someDatatype q;  
someDatatype *ptr = &q;
```



# Εφαρμογές

---

Ας εφαρμόσουμε επιτέλους τη θεωρία



# Επόμενη Λέξη

---

Το ζητούμενο είναι να γράψουμε μία 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.

Ως λέξη θεωρούμε κάθε ομάδα διαδοχικών αλφαβητικών χαρακτήρων, όπως τους θεωρεί και η `isalpha`

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)
```

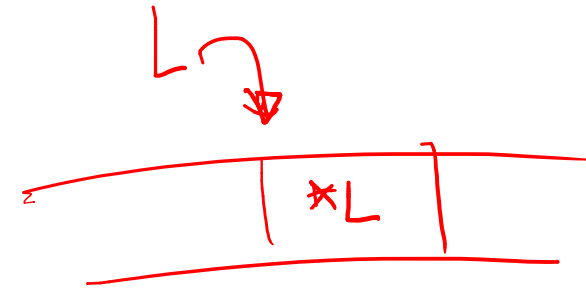
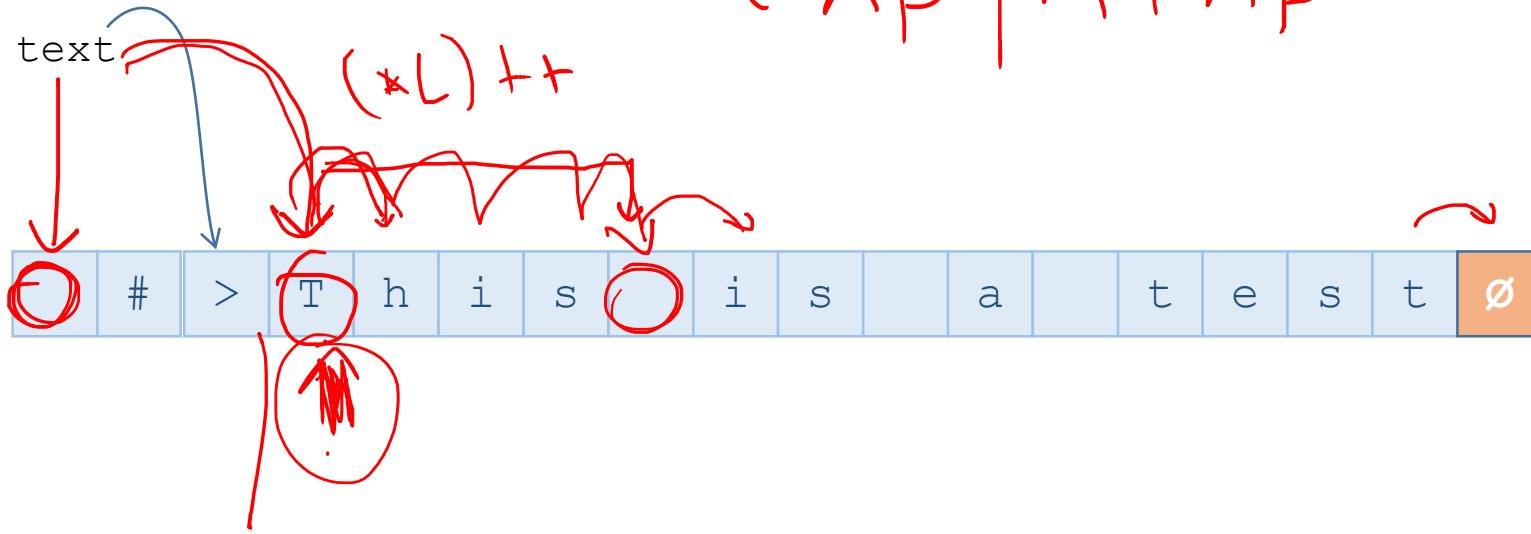
Η παραπάνω διαδικασία της σταδιακής συγγραφής της συνάρτησης απεικονίζει την ανάπτυξη της επίλυσης των προβλημάτων. Επιλύοντάς τα ένα προς ένα μπορούμε να ελέγχουμε την ορθότητα του κώδικά μας.

# Πρόχειρο

~~Ti Exw~~

a	T	D
a	a	&a
Ap	*Ap	Ap

isalpha ctype



$(*L)++$

~~$*L++$~~

# 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-23@allos.gr](mailto:c-programming-23@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.
- Τονίζουμε : Μην στείλετε **ποτέ κώδικα ως εικόνα**, είναι παντελώς άχρηστος!

