



# Επανάληψη για τις Τελικές εξετάσεις

(Διάλεξη 21)

Διδάσκων: Δημήτρης Ζεϊναλιπούρ

# Εισαγωγή



- Το μάθημα EPL032 έχει ως βασικό στόχο την **επίλυση προβλημάτων** πληροφορικής με την χρήση της **γλώσσας προγραμματισμού C**.
- Επομένως πρέπει:
  - Να κατανοήσετε τους διάφορους **κανόνες που διέπουν την γλώσσα C** και
  - Να **επιλύσετε διάφορα προγραμματιστικά προβλήματα** με χρήση αυτών των κανόνων.

# Επίλυση Προβλημάτων



- Πως επιλύουμε ένα πρόβλημα?
  1. Κατανόηση εκφώνησης από την περιγραφή του προβλήματος.
  2. Σχεδίαση της λύσης του προβλήματος (Εξεύρεση κατάλληλου Αλγόριθμου και σκιαγράφηση λύσης σε ψηλό επίπεδο )
  3. Κωδικοποίηση στην γλώσσα C.
  4. Έλεγχος και διόρθωση λαθών: Τρέχουμε (στο χαρτί) το πρόγραμμα για διάφορα στιγμιότυπα εισόδου.

# Τρόπος Μελέτης



- Για την επιτυχία σε αυτό το μάθημα χρειάζεται να κάνετε αρκετή εξάσκηση σε διάφορα προβλήματα
- Πως?
  - **Επανάληψη Σημειώσεων:** Σε κάθε εκφώνηση άσκησης, αποκρύψετε την λύση και προσπαθήστε να λύσετε μόνοι σας. Στην συνέχεια ελέγξτε εάν συμφωνεί η λύση σας με τις σημειώσεις. Αν βρήκατε διαφορετική λύση και δεν είστε σίγουροι εάν είναι σωστή δοκιμάστε την λύση σας στον υπολογιστή.
  - **Λύσεις Εργαστηριακών Ασκήσεων**  
Ξαναδείτε τις εργαστηριακές ασκήσεις μαζί με τις λύσεις (βρίσκονται στο Διαδίκτυο)
  - **Λύσεις Ασκήσεων Ενδιάμεσης Εξέτασης**  
Λύστε τις ασκήσεις της Ενδιάμεσης Εξέτασης (οι λύσεις βρίσκονται στο Διαδίκτυο)
  - **Παραδείγματα Ασκήσεων που δόθηκαν στο Εργαστήριο**

# Επανάληψη Σημειώσεων



## **Δ2: Εισαγωγή στον Προγραμματισμό (Κεφάλαιο 1.4-1.5)**

Κύκλος Ζωής Προγράμματος (Περιγραφή, Ανάλυση, Σχεδίαση Λύσης, Κωδικοποίηση, Έλεγχος, Συντήρηση)

## **Δ3-4: Εισαγωγή στην γλώσσα C (Κεφάλαιο 2.1-2.4)**

Σύνταξη, μεταβλητές και σταθερές (define),

## **Δ5-6: Τύποι Δεδομένων, Τελεστές & Αριθμητικές Εκφράσεις (Κεφ. 2.5)**

Βασικοί τύποι (char, int, float, double) και βασικοί τελεστές (+, -, \*, /, %, =), Μετατροπές Εκφράσεων στην C, Κανόνες Προτεραιότητας [ () \* / % + - = ], Αποτίμηση Εκφράσεων (π.χ.  $x * y * z + a / b - c * d$ ), Διαίρεση με 0, Μετατροπή Τύπων (αυτόματη π.χ. `float x = 5;` και ρητή `int x = (int) 5.0;`)

# Επανάληψη Σημειώσεων



## **Δ7: Είσοδος / Έξοδος Δεδομένων (Κεφάλαιο 2.6)**

Οι εντολές **printf**, **scanf** πάνω σε απλούς τύπους δεδομένων, **Ορίσματα** (%d %c %f %o %x %s), **Ειδικοί χαρακτήρες** (escape characters \n \t \\ ...), **Μορφοποίηση** Δεδομένων Εξόδου %n.md

## **Δ8: Συνθήκες Έλεγχου (Κεφάλαιο 4.1-4.9)**

Εντολές επιλογής (if-else if-else, switch), **Λογικές Παραστάσεις** (ΛΠ) (επιστρέφουν 0 ή 1). Δημιουργία ΛΠ με **Σχισιακούς Τελεστές** (<, >, ==, >=, <=, !=). Συνδυασμός ΛΠ με **Λογικούς Τελεστές** (&&, ||, !)

## **Δ9-10: Επαναληπτικοί Βρόχοι (Κεφάλαιο 5.1-5.3)**

**Δομές Επανάληψης:** while, for, do-while, Τελεστές Prefix & Postfix (++ , --), Σύνθετοι Τελεστές Ανάθεσης (-=, +=, /=, \*=, %=), Οι εντολές break/continue,

# Επανάληψη Σημειώσεων



## **Δ11-13: Συναρτήσεις (Κεφάλαιο 3.1-3.6)**

Δομημένος Προγραμματισμός (Αφαιρετικότητα), Συναρτήσεις Βιβλιοθήκης και Χρήστη (Πρότυπο, Ορισμός, Κλήση), Δομή Προγράμματος με συναρτήσεις, Συναρτήσεις & Παράμετροι, Εμβέλεια Μεταβλητών (Καθολικές & Τοπικές Μεταβλητές), Συναρτήσεις Βιβλιοθηκών `<stdio.h>`, `<stdlib.h>`, `<math.h>`,

## **Δ14: Αρχεία στην C (Κεφάλαιο 2.7-12)**

Δομή Αρχείων, Συναρτήσεις Επεξεργασίας Αρχείων στην C `fscanf`, `fopen`, `fprintf`, `fclose()`,

## **Δ15: Πίνακες (Κεφάλαιο 8.1-8.3)**

Εισαγωγή & Σύνταξη, Δήλωση Πίνακα, Αρχικοποίηση Πίνακα, Πρόσβαση, Παραδείγματα

## **Δ16: Παράλληλοι Πίνακες, Αλγόριθμοι Αναζήτησης & Ταξινόμησης (Κεφάλαιο 8.6)      LinearSearch, SelectionSort**

# Επανάληψη Σημειώσεων



## **Δ17: Πίνακες και Συναρτήσεις (Κεφάλαιο 8.4)**

Κλήση με τιμή και αναφορά, πίνακες και συναρτήσεις

## **Δ18: Πολυδιάστατοι Πίνακες (Κεφάλαιο 8.7)**

Πρότυπο - Ορισμός - Κλήση, Πίνακες 2 και 3 Διαστάσεων,  
Πολυδιάστατοι Πίνακες και Συναρτήσεις

## **Δ19-20: Αλφαριθμητικές Σειρές Χαρακτήρων (Κεφ. 9.3-9.4)**

Δήλωση, Αρχικοποίηση, Εκτύπωση, Ανάγνωση, Πίνακες  
από strings, συναρτήσεις βιβλιοθήκης <string.h>,  
Υλοποίηση συναρτήσεων βιβλιοθήκης Strings (strcat,  
strlen, strcmp, strcpy)



# Άσκηση 1 - **spaceSum**



- Η συνάρτηση **spaceSum** επιστρέφει τον αριθμό κενών (space) σε μια συμβολοσειρά απροσδιορίστου μεγέθους.

**Πρότυπο Συνάρτησης:** `int spaceSum(char str[]);`

**Παράδειγμα Εκτέλεσης:**

```
char str[]="University of Cyprus";
```

```
printf("%d", spaceSum(str));
```

**ΕΚΤΥΠΩΝΕΙ 2**

# Άσκηση 1 - spaceSum



```
int spaceSum(char str[])
{
    int sum = 0;
    int i = 0;

    while (str[i] != '\0') {
        if (str[i] == ' ') { sum++; }
        i++;
    }

    return sum;
}
```

# Άσκηση 2 - mystrstr



- Η συνάρτηση **mystrstr** ψάχνει να βρει την πρώτη εμφάνιση ενός string **b[]** σε ένα string **a[]**. Εάν το **b** βρεθεί μέσα στο **a** τότε η **mystrstr** επιστρέφει την θέση στην οποία βρέθηκε το **b**. Στην αντίθετη περίπτωση επιστρέφει **-1**.

**Πρότυπο Συνάρτησης:** `int mystrstr(char a[], char b[]);`

**Παράδειγμα Εκτέλεσης:**

```
char a[]="University";
```

```
char b[]="ver";
```

```
int pos = mystrstr(str);
```

```
printf("%d", pos);
```

**a**

0	1	2	3	4	5	6	7	8	9	10
U	n	i	v	e	r	s	i	t	y	\0

**b**

0	1	2	3
v	e	r	\0

**pos**

**ΕΚΤΥΠΩΝΕΙ 3**

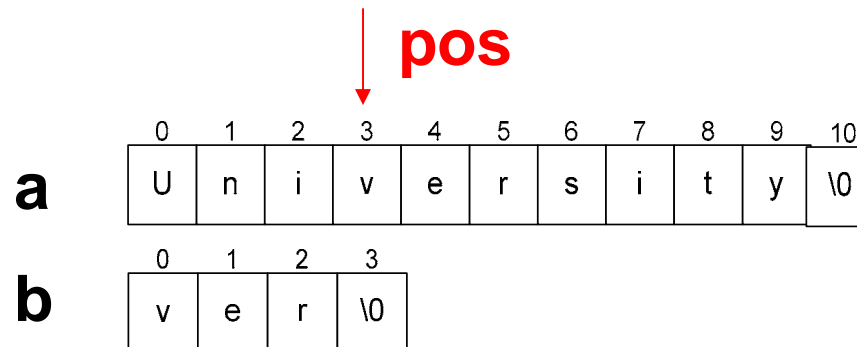
# Άσκηση 2 - mystostr



- **Περιγραφή του αλγόριθμου με λόγια**

Διαβάζουμε σειριακά την συμβολοσειρά  $a$ , μέχρι να βρούμε κάποιο στοιχείο στην θέση  $i$ , για το οποίο ισχύει  $a[i]==b[0]$  (η θέση  $i$  σημειώνεται στην μεταβλητή  $pos$ ). Στην συνέχεια ελέγχουμε εάν η ισότητα ισχύει και για όλους τους χαρακτήρες του  $b$ , δηλαδή :  
 $a[i+1]==b[1]$ ,  $a[i+2]==b[2]$ , ... κτλ.

Εάν φτάσουμε με τον πιο πάνω τρόπο στο τέλος της συμβολοσειράς  $b$ , τότε επιστρέφουμε το  $pos$ . Στην αντίθετη περίπτωση επιστρέφουμε  $-1$ .



# Άσκηση 2 – strstr με 2 while loops



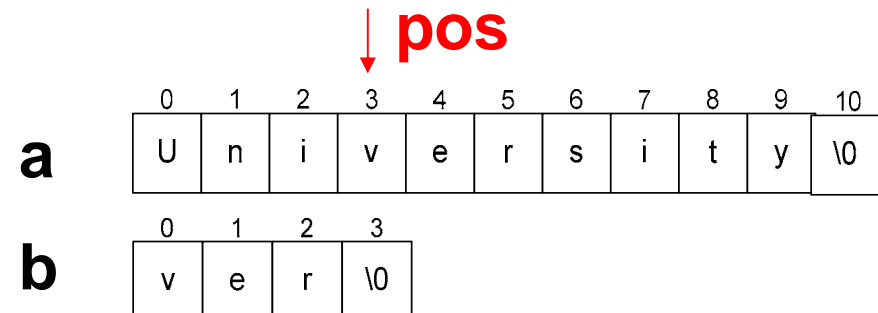
```
int mystostr(char a[], char b[])
{
    int i = 0, j = 0;
    int pos=-1;
    while (a[i] != '\0') {
        pos = i;
        while (a[i] == b[j]) {
            j++; i++;

            // Εάν ο επόμενος b[j] είναι ο NULL τότε έγινε fullmatch!
            if (b[j] == '\0') return pos;

            // Εάν ο a έφτασε στο τέλος (αλλα όχι ο b) π.χ a="unive" b="ver"
            if (a[i] == '\0') return -1;
        }
        i++;
    }
    return -1;
}
```

Η μεταβλητή j μετρά  
πάνω στο b[], ενώ η i  
πάνω στο a[].

// Σημειώνουμε την αρχική θέση  
// Το matching ξεκινά

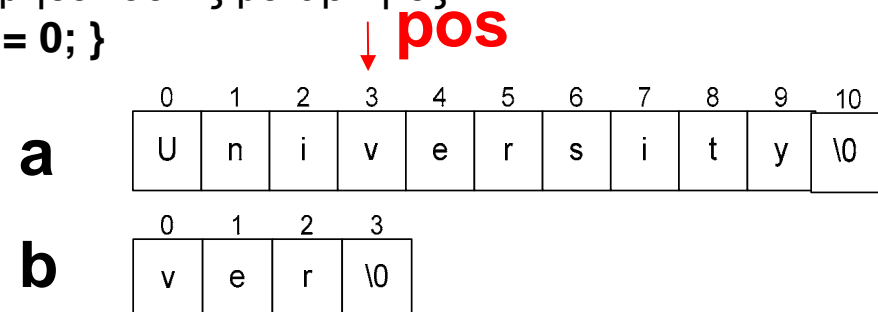


# Άσκηση 2 – mystrstr με 1 while loop

```
int mystrstr(char a[], char b[])
{
    int i = 0, j = 0; pos = -1;

    while (a[i] != '\0')
    {
        // Το matching ξεκινά
        if (a[i] == b[j]) {
            // Σημειώνουμε την θέση στην οποία ξεκίνησε το match
            if (j == 0) { pos = i; }
            j++;
            // Εάν ο επόμενος είναι ο NULL τότε έγινε fullmatch
            if (b[j] == '\0') { return pos; }
        }
        else { // Εάν το matching ξεκίνησε αλλά δεν ταιριάζουν
            // οι παρόν χαρακτήρες τότε μηδένισε τις μεταβλητές.
            if (pos > -1) { pos = -1; j = 0; }
        }
        i++;
    }
    return -1;
}
```

Η μεταβλητή j μετρά  
πάνω στο b[], ενώ η i  
πάνω στο a[].



# Άσκηση 3 - delChar



- Η συνάρτηση **delChar** διαγράφει κάθε εμφάνιση κάποιου χαρακτήρα x από την συμβολοσειρά a[].

**Πρότυπο Συνάρτησης:** int delChar(char a[], char x);

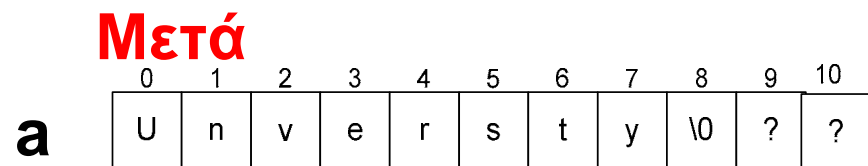
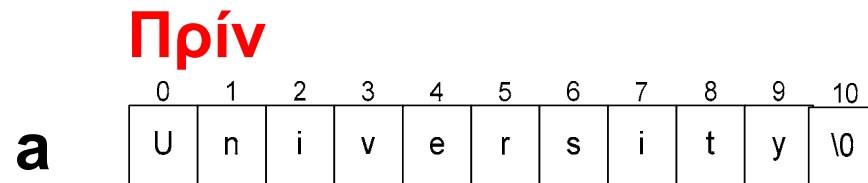
**Παράδειγμα Εκτέλεσης:**

```
char a[]="University";
```

```
char x='i';
```

```
delchar(a, x);
```

```
printf("%s", a);
```



**Εκτυπώνει «Unversty»**

## Άσκηση 2 - delChar



### Περιγραφή του αλγόριθμου με λόγια

1. Διαβάζουμε σειριακά την συμβολοσειρά  $a[]$ . Σε κάθε ανάγνωση κάποιου χαρακτήρα του  $a[]$ , ελέγχουμε εάν τον χρειαζόμαστε ή όχι.
2. Αν τον χρειαζόμαστε τότε τον αντιγράφουμε στην θέση  $j$  (αρχικά  $j=0$ ) και αυξάνουμε το  $j++$ .
3. Επαναλαμβάνουμε τα βήματα 1-2 μέχρι να φτάσουμε στο τέλος της συμβολοσειράς  $a[]$ .



# Άσκηση 3 – delChar



```
void delChar(char a[], char x)
```

```
{
```

```
    int i = 0;    // θέση στο string a
```

```
    int j = 0;
```

```
    while (a[i] != '\0') {
```

```
        // Εάν ο χαρακτήρας είναι διαφορετικός από τον x
```

```
        // τότε αντίγραψε τον στην θέση a[j]
```

```
        if (a[i] != x) {
```

```
            a[j] = a[i];
```

```
            j++;
```

```
        }
```

```
        i++;
```

```
    }
```

```
    a[j] = '\0';
```

```
}
```

**Πρίν**

	0	1	2	3	4	5	6	7	8	9	10
<b>a</b>	U	n	i	v	e	r	s	i	t	y	\0

**Μετά**

	0	1	2	3	4	5	6	7	8	9	10
<b>a</b>	U	n	v	e	r	s	t	y	\0	?	?

# Πληροφορίες Εξέτασης



**Μέρα:** Πέμπτη, 11 Μαΐου

**Ώρα:** 8:30-11:30

**Τοποθεσία:** Πανεπιστημιούπολη

Ακροατήριο 1 - ΧΩΔ103 (Επίθετα Α-Μιχαηλίδου)

Ακροατήριο 1 - ΧΩΔ104 (Επίθετα Μωησέως-Ω)

Ακροατήριο 2 - ΧΩΔ110

**Ποσοστό Τελικής:** 50%

Ποσοστό Ενδιάμεσης 20%

Ποσοστό Εργαστηρίων: 30%

**Καλή επιτυχία!**