Δομές struct και pointers

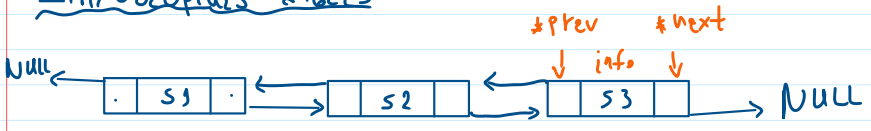
```
// Δήλωση μιας δομής struct
struct mystruct {
    int a;
    char b[5];
    float f;
};
// Δήλωση μιας μεταβλητής struct τύπου mystruct
int main () {
    struct mystuct mystruct1;
    // Αρχικοποίηση τιμών
    mystruct1.a = 30;
    mystruct1.f = 3.14;
    strcpy(mystruct1.b, "abcd");

// Δήλωση struct pointer τύπου mystruct
    struct mystruct *ptr;
    ptr = &mystruct1;
// Αρχικοποίηση τιμών με τον χρήση pointer
    ptr -> a = 30;   ή  (*ptr).a = 30;
    ptr -> f = 3.14;
    strcpy (ptr->b, "abcd")
}
```

ptr=>Δ.Μ. mystruct1
*ptr == mystruct1

Λίστες

-Διπλοσυνδεδεμένες λίστες




```
struct double {
    struct double *prev;
    int info;
    struct double *next;
};
```

- Μονοδεσμικές Λίστες


```
struct single {
    int info;
    struct double *next;
};
```

Άσκηση:

4. (15%) Έστω η δήλωση

```
struct card {
    int info;
    struct card *prev;
    struct card *next;
};
struct card *first, *last, *temp;
```

Και έστω ότι, μετά την εκτέλεση κάποιων εντολών, μια λίστα έχει την ακόλουθη μορφή στη μνήμη:

	Όνομα Πεδίου ή Μεταβλητής	Περιεχόμενο	Θέση Μνήμης
	...		
s1	info	500	2500
	prev	5500	
	next	7500	
	...		
	first	6500 (α)?	3500
	...		
	last	7500	4500
	...		
s2	info	800	5500
	prev	6500 (β)?	
	next	2500 (γ)?	
	...		
s3	info	400	6500
	prev	NULL	
	next	5500 (δ)?	
	...		
s4	info	300	7500
	prev	2500	
	next	NULL (ε)?	

4.1. (5%) Υποθέτοντας ότι η λίστα έχει δημιουργηθεί σωστά, συμπληρώστε τον παραπάνω πίνακα στις απαντήσεις σας, αντικαθιστώντας τα (α)?,(β)?,(γ)?,(δ)?,(ε)?. NULL ⟵ s3 ⟷ s2 ⟷ s1 ⟷ s4 ⟶ NULL

4.2. (4%) Δείξτε τι θα εμφανίσουν οι παρακάτω εντολές:

4.2.1. `printf("%d",&last);` 4500

4.2.2. `printf("%d",last);` 7500

4.2.3. `printf("%d",last->prev);` Πρόσβαση στο πεδίο prev του struct obj που δείχνει ο pointer last (2500)

4.2.4. `printf("%d",last->prev->info);` last/s4/s1.info (500)

4.3. (6%) Έστω ότι ο temp "δείχνει" σε έναν ανεξάρτητο κόμβο εκτός λίστας. Δείξτε την ακολουθία εντολών που ελέγχει αν η τιμή που περιέχει το info του ανεξάρτητου κόμβου υπάρχει στη λίστα και εμφανίζει "Found!" αν το βρει (η λίστα μπορεί να είναι κενή). Υποθέστε ότι γνωρίζετε τα πεδία του struct και γνωρίζετε ότι οι first, last (πρέπει να) «δείχνουν» στον πρώτο και τον τελευταίο κόμβο της λίστας (αν υπάρχουν).

```
struct card *current = NULL;
if (!first && !last) {
        printf("not found");
}
else {
    current = first;
    while (current) {
        if (current->info == temp->info) {
            printf ("Found!");
            break;
        }
        current = current->next;
    }
}
```

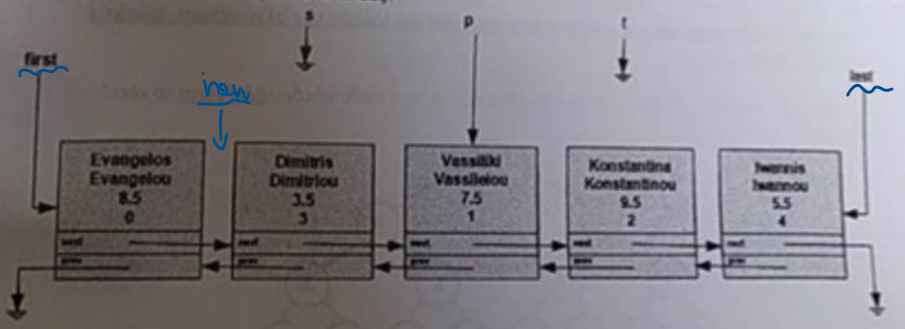
Άσκηση :

5. (20%) Έστω η δήλωση

```
struct card {
    char onom[15];
    char epwn[15];
    float bath;
    int a;
    struct card *next;
    struct card *prev;
};

struct card *p,*s,*t,*first,*last;
```

Και έστω ότι, μετά την εκτέλεση κάποιων εντολών, μια λίστα έχει την ακόλουθη μορφή (στο παρακάτω σχήμα το ↓ υποδηλώνει ότι ένας δείκτης είναι NULL και τα πεδία εμφανίζονται μέσα στους κόμβους με τη σειρά που δηλώθηκαν στο struct):



5.1. (4%) Γράψτε την έκφραση που αντιπροσωπεύει τη διαφορά των βαθμών μεταξύ του 1ου και του τελευταίου κόμβου της λίστας χρησιμοποιώντας τους υπάρχοντες pointers.

5.2. (8%) Γράψτε τις εντολές για να δημιουργηθεί ένας νέος κόμβος με περιεχόμενο Manos Manou με βαθμό 7.5 και 2 απουσίες και να εισαχθεί μεταξύ 1ου και 2ου κόμβου, χρησιμοποιώντας τους υπάρχοντες pointers.

5.3. (8%) Γράψτε τις εντολές για την εύρεση του ελάχιστου αριθμού απουσιών μέσα στη λίστα, έτσι ώστε οι εντολές να λειτουργούν ανεξάρτητα από το πλήθος των κόμβων της λίστας (μπορεί να είναι και κενή). Υποθέστε ότι γνωρίζετε τα πεδία του struct και γνωρίζετε ότι ο first «δείχνει» στον 1ο κόμβο της λίστας (αν υπάρχει).

5.1) first → bath - last → bath

5.2) struct card new;

```
strcpy(new.onom, "Manos");
strcpy(new.epon, "Manou");
new.bath = 7.5;
new.a = 2;

new.next → (first → next)
new.prev → first
```