

Δομές Δεδομένων, Εξεταστική περίοδος Σεπτεμβρίου 2021
Ενδεικτικές λύσεις θεμάτων

- 1) Ας υποθεθεί ότι υπάρχει η λίστα L με δομή

```
struct Lista {  
    int code; //κωδικός  
    int priority; // προτεραιότητα  
    char onmo[40]; //ονοματεπώνυμο  
    struct lista *epomeno; }
```

Η λίστα περιέχει «πελάτες» οι οποίοι έχουν σειρά προτεραιότητας priority και είναι τοποθετημένοι τυχαία στην λίστα ανεξαρτήτου προτεραιότητας. Εάν είναι γνωστό ότι το μέγεθος της λίστα είναι N και επομένως το πεδίο priority περιέχει τιμές από 1 μέχρι N (αλλά σε τυχαία θέση μέσα στην λίστα) γράψτε τον κώδικα που να μεταφέρει την λίστα σε ουρά τοποθετώντας τους πελάτες στην σωστή θέση. Το πεδίο priority προφανώς δεν χρειάζεται στην ουρά. **Δηλώστε όλες τις μεταβλητές/δομές που θα χρησιμοποιήσετε.** Παρατήρηση: Υποτίθεται ότι η συνάρτηση void eisagogi_oura(Oura *x) υπάρχει και δεν χρειάζεται να την γράψετε αλλά μπορείτε να την χρησιμοποιήσετε για εισαγωγή ενός στοιχείου στην ουρά. (5 μονάδες)

Αν και παρακάτω παρατίθεται το πλήρες πρόγραμμα, μόνο ότι είναι με έντονη γραφή θα έπρεπε να γράψετε.

```
#include <stdio.h>  
#include <string.h>  
typedef struct Lista {  
    int code; //κωδικός  
    int priority; // προτεραιότητα  
    char onmo[40]; //ονοματεπώνυμο  
    struct Lista *epomeno; };
```

```
typedef struct Oura {  
    int code; //κωδικός  
    char onmo[40]; //ονοματεπώνυμο  
    struct Oura *epomeno; };
```

```
Lista *kefali_lista=NULL, *oura_lista=NULL;  
Oura *kefali_oura=NULL, *oura_oura=NULL;
```

```
void eisagogi_arxi(Lista *);  
void emfanisi_listas(void);  
void eisagogi_oura(Oura *);  
void provoli_ouras(void);
```

```

int main() {
    Lista L, *x;
    Oura O;
    int N=5; //πλήθος ατόμων στην λίστα και ουρά
    int i;
    int temp_priority=1;

    // εισαγωγή στην λίστα
    for (i=1; i<=N; i++) {
        printf("Κωδικός:");
        scanf("%d", &L.code);
        printf("\nΟνοματεπώνυμο:");
        scanf("%s",&L.onmo);
        printf("\nΠροτεραιότητα:");
        scanf("%d", &L.priority);
        eisagogi_arxi(&L);
    }
    emfanisi_listas();

    do {
        x = kefali_lista;
        for (i=1; i<=N; i++) {
            if (temp_priority == x->priority) {
                O.code=x->code;
                strcpy(O.onmo, x->onmo);
                eisagogi_oura(&O);
            }
            x = x->epomeno;
        }
        temp_priority++;

    } while (temp_priority<=N);

    provoli_ouras();

    return 0;
}

void eisagogi_arxi(Lista *p)
{
    Lista *neo;
    neo = (Lista *) malloc(sizeof(Lista));

```

```

*neo = *p;
if ( kefali_lista == NULL) {
    kefali_lista = neo;
    oura_lista = neo;
}
else {
    neo->epomeno = kefali_lista;
    kefali_lista = neo;
}
}

void emfanisi_listas(void)
{
    Lista *x;
    int c = 0;

    x = kefali_lista;
    while (x != NULL) {
        printf("\n%5d %40s %5d", x->code, x->onmo, x->priority);
        x = x->epomeno;
    }
    printf("\n\n");
}

void eisagogi_oura(Oura *x)
{
    Oura *neos_konvos;

    neos_konvos = (Oura *) malloc(sizeof(Oura));
    if ( neos_konvos == NULL )
        printf("\n\n Αδύνατη δέσμευση μνήμης. Η εισαγωγή δεν έγινε!\n\n");
    else {
        *neos_konvos = *x;
        neos_konvos->epomeno = NULL;
        if ( kefali_oura == NULL) {
            kefali_oura = neos_konvos;
            oura_oura = neos_konvos;
        }
        else {
            oura_oura->epomeno = neos_konvos;
            oura_oura = neos_konvos;
        }
    }
}
}

```

```

void provoli_ouras(void) {
    Oura *x, *x_epomeno;

    x = kefali_oura;
    while ( x != NULL) {
        printf("\n==> %d, %s", x->code, x->onmo);
        x_epomeno = x->epomeno;
        x = x_epomeno;
    }
    printf("\n\n");
}

```

- 2) Εάν υπάρχει το αρχείο κειμένου "Olympics_2020.txt" όπου σε κάθε γραμμή εμφανίζεται η πληροφορία <Κωδικός χώρας, Χώρα, χρυσά μετάλλια, αργυρά μετάλλια, χάλκινα μετάλλια>. Γράψτε εκείνο το κομμάτι του κώδικα που θα μεταφέρει το αρχείο σε ένα αρχείο εγγραφών "Olympics_2020.dat" και στο τέλος να εκτυπώνει στην οθόνη α) το πλήθος των χωρών, β) το πλήθος όλων των μεταλλίων που έχουν δοθεί (αθροιστικά, ανεξάρτητα είδους) (5 μονάδες).

Αν και παρακάτω παρατίθεται το πλήρες πρόγραμμα, μόνο ότι είναι με έντονη γραφή θα έπρεπε να γράψετε.

```

#include <stdio.h>
struct Countries {
    int code;
    char onoma[30];
    int gold;
    int silver;
    int bronze;
};
int main() {
    FILE *S, *T; //αρχεία κειμένου - εγγραφών
    struct Countries C;
    int counter_countries = 0, counter_medals = 0;
    int i;

    S = fopen("Olympics_2020.txt", "r");
    T = fopen("Olympics__2020.dat", "wb");

    while (! feof(S) ) {
        fscanf(S, "%d%s%d%d%d", &C.code, &C.onoma, &C.gold, &C.silver, &C.bronze);
        if ( ! feof(S) ) {
            fwrite(&C, 1, sizeof(C), T);

```

```

        counter_countries++;
        counter_medals += C.gold+C.silver+C.bronze;
        printf("%5d %30s %5d %5d %5d\n",C.code, C.onoma, C.gold, C.silver,
C.bronze);
    }
}
for (i=1; i<=54; i++)
    printf("-");
printf("\n%36d%18d\n\n", counter_countries, counter_medals);
printf("\n\nΣύνολο χωρών: %d", counter_countries);
printf("\n\nΣύνολο μεταλλίων: %d\n\n", counter_medals);

fclose(S);
fclose(T);

return 0;
}

```