

# Lab#2

## Algortem & programim i avancuar

A. GOGU

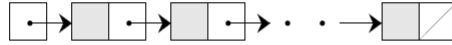
### 1 Objectiv

Qellimi i ketij laboratorii eshte familjarizimi me veprimet me te perdorshme te listave te lidhura (krijimi i listes, shtimi i njeve, fshirja e njeve, kerkimi i informacionit) dhe ndertimin e nje aplikacioni qe perdor listat zinxhir.

### 2 Kodimi

Listat zinxhir jane struktura te dhenash me aksesim indirekt. Ato lejojne ndertimin e nje vargu elementesh te te njejtit tip (shume rralle te tipeve te ndryshme). Avantazhi i tyre eshte dinamizmi. Ne fakt, ne alokojme memorie per cdo element te listes dhe e menaxhojme veprimet ne funksion te ndryshimeve te madhesis se listes. Si kategori dallojme: listat nje fishe, dy fishe dhe rrethore. Per te thjeshtezuar algoritmet, kryesisht perdorim listat rrethore ne te cilat kemi vetem nje nyje qe tregon fillimin dhe fundin.





### 3 Aplikacioni

Ne kete lab, ju duhet te ndertoni nje aplikacion te thjeshtezuar per menaxhimin e stokut ne magazina. Stoku eshe i perbere nga nje numer artikujsh. Cdo artikull ka si minimum keto te dhena:

- numrin unik te identifikimit
- emrin e artikullit ne magazine
- cmimin perkates
- sasine perkatese
- kategorine (ushqim, pije, higjene, tekstile, artikull guxhine, etj.)

Funksioni baze ne kete aplikacion eshte dergimi i kerkesave (Requests) per stokun qe lidhet:

- numrin identifikues te artikullit
- sasine qe duhet te shtohet/hiqet per artikullin.

### 4 Struktura e te dhenave

Ne kete aplikacion, ju duhet te respektoni domosdoshmerisht keto struktura te dhenash:

1. Struktura per te ruajtuar te dhenat e artikullit:

```
typedef struct Article
{
    int id; // numrin identifikues te artikullit
    char name[50]; // emri i artikullit
    float PU; // cmimi ne cope apo njesi
    int quantity; // sasia ne stok e artikullit
    char category[50]; //(ushqim, pije, higjene, tekstile, etj.)
```

```

        struct Article * nextArticle;
    };
    typedef Article * ListeArticles;

```

2. Struktura per te ruajtuar te dhenat e kerkesave:

```

typedef struct Request
{
    int idArticle; // numri identifikues i artikullit
    int qte; // sasia qe i korresponndon artikullit me id e dhene
    struct Request * nextRequest;
};
typedef Request * ListeRequests;

```

Ne kete baze te dhenash duhet te jene tre lista:

- *Liste\_stock\_dispo* : permban artikujt e disponueshem
- *Liste\_stock\_finished* : permban artikujt qe kane mbaruar
- *Liste\_Req* : te gjitha perditsimet e bera mbi artikujt.

## 5 Devlopimi i funksioneve

Funksionet qe duhen shkruar per te menaxhuar magazinën janë:

1. Funksioni i cili shton nje artikull ne nje pozicion te tille ne liste qe cmimi per njesi te jete ne rend zbrites.

```
void AddArticle(ListeArticles l, Article * a)
```

2. Funksioni i cili afishon listen sic tregohet ne tabelen 1.

```
void DisplayStock(ListeArticles l)
```

| Id number | name       | Price/unit | Stock quantity | category     |
|-----------|------------|------------|----------------|--------------|
| 15        | Pen        | 2,50       | 25             | school items |
| 20        | Math Books | 2,50       | 100            | school items |

Table 1: Afishimi

3. Funksioni qe fshin te gjithe artikujt ne liste, pa asnje perjashtim.

```
void DeleteStock(ListeArticles l)
```

4. Funksioni qe fshin artikullin me numer identifikimi qe jepet nga *id\_Article*.

```
void DeleteArticle(ListeArticles l, int id_Article)
```

5. Funksioni:

```
ListeArticles ResearchArticleOnCriteria(ListeArticles l, int id,
char* name, char* category)
```

ky funksion shfaq ne ekran listen e artikujve sipas kritereve te kaluara si argument. Keto kritere jane: numri identifikues, emri, dhe/ose kategoria. Nese perdoruesi kerkon te shmang kerkimin sipas *id*, atehere ai duhet te kaloj si parameter vleren  $-1$ , dhe per parametrat e tjere stringen "IGNORE". Shembull:

```
ResearchArticleOnCriteria(Liste_stock_dispo, 12, "pen", "school items")
```

Funksioni duhet te shfaqe vetem artikullin perkates me *id* = 12, name pen, dhe category school items.

```
ResearchArticleOnCriteria(Liste_stock_dispo, -1, "IGNORE", "school items")
```

Funksioni duhet te shfaqe vetem artikullin perkates me category= school items.

6. Funksioni qe shfaq te gjithe artikujt sipas sasise te percaktuar nga kufiri minimal dhe ai maksimal [*qte\_min*, *qte\_max*].

```
ListeArticles ResearchOnQuantity(Liste_stock_dispo, int qte_min, int qte_max)
```

7. Funksioni qe shfaq te gjithe artikujt sipas sasise te percaktuar nga kufiri minimal dhe ai maksimal: [*price\_min*, *price\_max*].

```
ListeArticles ResearchOnPrize (Liste_stock_dispo, int price_min, int price_max)
```

8. Funksioni qe perditeson sasine e nje artikulli te dhene sipas numrit identifikues. Nese sasia e artikullit behet 0, atehere artikulli duhet te hqiet nga lista *Liste\_stock\_dispo* dhe te shtohet tek *Liste\_stock\_finished*.

```
void UpdateArticle(int id, int quantity)
```

,

9. Funksionin, i cili merr si parameter nje list artikujsh dhe zbret nga gjendja sasine perkatese te tyre.

```
void SaleArticles(ListeRequetes r)
```

*Si argument kalohet lista e kerkesave.*

10. Funksioni:

```
void Furniturep(ListeRequetes r)
```

i cili merr si parameter nje list artikujsh dhe shton tek gjendja sasine perkatese te tyre. *Si argument kalohet lista e kerkesave.*

*Kujdes! Funksionet 9 dhe 10 therrasin funksionin UpdateArticle().*

*Kujdes! File qe keni marr permбан te dhena qe nuk jane te renditura. Cdo rresht permбан te dhenat e nje artikulli: Id number, name, Price/unit, Stock quantity dhe category, ta ndara me TAB.*

## 6 Nderfaqja

Perdoruesi duhet te kete nje nderfaqe ne menyre te tille qe te njoh ne cdo moment gjendjen e stokut, sasine qe ngelet ne nje kategori, sasine qe planifikohet te blihet, dhe gjithashtu veprime te tjera te bazuara ne funksionet me lart: Shkruani programin qe nderton menune e meposhtme:

1. Leximi i te dhenave nga nje file
2. Shtimi i nje artikulli te ri nga tastiera
3. Afishimi i karateristikave te stokut
  - (a) Stoku i diponueshem
  - (b) Stoku i mbaruar.
  - (c) i gjithe stoku
4. Fshirja e nje artikulli
5. Kerkimi i nje artikulli

- (a) kerkimi sipas emrit, id, dhe/ose kategorise
  - (b) kerkimi sipas sasise
  - (c) kerkimi sipas cmimit
6. Menaxhimi i kopmandave te klientit
  7. Furnizimi
  8. Fshirja e stokut
  9. Dalja nga programi.

Shenim: Programi duhet te permbaje 3 file: `tp3.h` : permban konstantet globale, tipet dhe prototipet e funksioneve. `tp3.c` :permban funksionet e zhvilluara. `tp3_main.c` : permban vetem `main()`. Te tjere funksione mund te fusni per funksionimin me te mire te kerkesave. Ju ngelet per detyre t'i identifikoni dhe t'i zhvilloni. Ju duhet te dorezoni nje raport 4 faqesh qe do te perfshinte: hyrjen, zgjedhjet e bera ne programim, llogaritjen e kompleksitetit, perfundimet, pa harruar edhe trajtimin e problemeve specifike qe keni hasur, zgjidhjet perkatese, dhe funksionet e shtuara. Kodi burim dhe raporti duhet te dergohen me email brenda 2-javeshe.