

REPUBLIQUE ALGERIEN DEMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR



**Université des Sciences et de la Technologie
Houari Boumediene**

**Rapport
Projet TP ASD**

Réalisé par :

Ferradji Younes

212131036889

Mabed Hichem

191931046444

Type Date

Les attributs:

- **jj**: jour de type entier
- **mm**: mois de type entier
- **aaaa**: année de type entier

Les fonctions du type Date :

- **DateCMP** : cette fonction fait la comparaison entre deux date et retourne un entier et se dépend de résultat de cet entier la comparaison sera faite.

Déclaration du type Date

typedef struct Date Date;

```
struct Date
{
    int jj, mm, aaaa;
};
```

Pour utiliser dans les autres structures comme la date de péremption, la date de la commande ...etc.

Type Article

Les attributs :

- **QTC** : la quantité de cet article dans le stock, de type entier.
- **Designation** : la référence de cet article pour la recherche et autre traitement, type entier.
- **Prix** : le prix d'unité de l'article, type réel.
- **Critique** : si l'article est critique alors la valeur sera à 1 sinon 0, type entier.
- **Date_per** : la date de péremption, type Date.
- **NBvente** : nombre des unités vendu de cet article, type entier.
- **Suiv** : pointeur qui pointe à l'élément suivant pour faire le chainage, type Article.

Les fonctions du type Article :

- **EmpilerArti** : cette procédure marche comme empiler dans une pile tout simplement, on la donne dans les paramètres la Pile des articles et l'article à empiler, et elle va l'empiler dans la Pile.
- **AjouArti** : cette procédure reçoit en paramètres la Pile des articles, alloue un espace du type article, après elle demande aux utilisateurs d'introduire les données du nouveau article à ajouter qui seront scanner directement dans les attributs du nouveau élément allouer, et après elle empiler ce nouveau article.
- **DepilerArticle** : une fonction de type de retourne Article, qui marche comme dépiler où elle va dépiler le sommet de la pile et le renvoi dans le retourne.
- **ModifierArt** : une procédure qui reçoit en paramètres la Pile des articles, demande de l'utilisateur d'introduire la désignation de l'article qu'il veut le faire un changement, cette procédure va lancer une recherche de cet article, s'il existe alors un menu sera afficher contenant les changements possible dans les attributs de cet article, et l'utilisateur doit choisir une option et faire le changement nécessaire, sinon si l'article n'existe pas alors un message sera afficher que l'article n'existe pas.
- **SupprimerArticle** : une procédure qui reçoit en paramètres la Pile des articles, demande de l'utilisateur d'introduire la désignation de l'article qu'il veut le supprimer, elle fait la recherche de cet article en dépiler, test, empiler dans une pile temporaire jusqu'à que l'article sera trouvé, si l'article est trouvé alors il ne sera pas empiler dans pile temporaire, et la boucle de recherche arrête, la boucle de reconstruction commence où on dépile de la pile temporaire et on empile dans la Pile initial.
- **affArtiList** : une procédure qui reçoit en paramètres la Pile des articles, dépile le sommet de cette pile, faire l'affichage des attributs, empile dans une pile temporaire, et après quand la pile est vide, elle fait la reconstruction où elle dépile de la pile temporaire et empile dans la pile initial.
- **AffArtCrit** : une procédure qui reçoit en paramètres la Pile des articles, dépile le sommet de cette pile, test si cet article est critique alors faire l'affichage des attributs, empile dans une pile temporaire, et après quand la pile est vide, elle fait la reconstruction où elle dépile de la pile temporaire et empile dans la pile initial.

- **Rech_carac** : une procédure qui reçoit en paramètres la Pile des articles et un caractère pour savoir quel option de recherche l'utilisateur veut faire sa recherche, elle test la valeur du caractère et selon sa elle demande de donnée la valeur de l'attribut rechercher et elle fait la recherche, si l'article est trouvé elle l'affiche, sinon elle affiche un message que l'article n'existe pas.
- **Affiche_arti_exp** : une procédure qui reçoit en paramètres la Pile des articles, elle dépile le sommet de cette pile, test si cet article expirer en utilisant la fonction Date_Cmp en comparent la date de péremption de l'article avec la date courante, alors faire l'affichage des attributs, empile dans une pile temporaire, et après quand la pile est vide, elle fait la reconstruction où elle dépile de la pile temporaire et empile dans la pile initial.

Déclaration du type Article

typedef struct Article Article;

```
struct Article
{
    int QTC;
    char Designation[20];
    float Prix;
    int Critique; // si 1 qlors oui sinon 0 alors non
    Date date_per;
    int NBvente;
    Article *suiv;
};
```

Déclaration du la Pile d'Articles

typedef struct Aliste Aliste;

```
struct Aliste
{
    Article *Sommet;
};
```

Type Personne

Les attributs :

- **ID** : identifiant de chaque personne, type entier.
- **Nom** : nom de la personne, type chaîne de caractère.
- **Prenom** : prénom de la personne, type chaîne de caractère.
- **Phone** : numéro du téléphone de la personne, type entier.
- **Adresse** : adresse de la personne, type chaîne de caractère.
- **Etat** : pour savoir si cette personne est un client(1) ou fournisseur(2) ou divers(1), type entier.
- **NBCom** : nombres des commandes faites par cette personne, type entier.
- **Suiv** : pointeur qui pointe à l'élément suivant pour faire le chaînage, type Personne.

Les fonctions du type Personne :

- **AjouPer** : cette procédure reçoit en paramètres la File des Personnes, alloue un espace du type Personne, après elle demande aux utilisateurs d'introduire les données du nouvel personne à ajouter qui seront scanner directement dans les attributs du nouveau élément allouer, et après elle enfile cette nouvelle personne.
- **EnfilerPer** : cette procédure marche comme enfile dans une file tout simplement, on la donne dans les paramètres la file des personnes et la personne à enfile, et elle va l'enfile dans la file.
- **DefilerPer** : une fonction de type de retourne Personne, qui marche comme défile où elle va dépiler la tête de la file et le renvoi dans le retourne.
- **SupprimerPersonne** : une procédure qui reçoit en paramètres la File des Personnes, demande de l'utilisateur d'introduire l'ID de la personne qu'il veut la supprimer, elle fait la recherche de cet article en défile, test, enfile dans une file temporaire jusqu'à que la personne sera trouvé, si la personne est trouvé alors il ne sera pas enfile dans file temporaire, et la boucle de recherche arrête, la boucle de reconstruction commence où on défile de la file temporaire et on enfile dans la File initial.
- **ModifierPer** : une procédure qui reçoit en paramètres la File des Personnes, demande de l'utilisateur d'introduire l'ID de la personne qu'il veut la faire un changement, cette procédure va lancer une recherche de cette personne, si elle existe alors un menu sera afficher contenant les changement possible dans les attributs de cette personne, et l'utilisateur doit choisir une option et faire le changement nécessaire, sinon si la personne n'existe pas alors un message sera afficher que la personne n'existe pas.

Déclaration du type Personne

typedef struct Personne Personne;

```
struct Personne
{
    int ID; // numero client
    char Nom[15];
    char Prenom[15];
    int Phone;
    char Adresse[30];
    int Etat
    int NBCom;
    Personne *suiv;
};
```

Déclaration du File des Personnes

typedef struct Pliste Pliste;

```
struct Pliste
{
    Personne *TeteP;
    Personne *QueueP;
};
```

Type Commande

Les attributs :

- **IDclient** : identifiant du personne qui a fait la commande, type entier.
- **Type** : le type de la commande, si 1 alors vente sinon si 0 alors achat, type entier.
- **NFacture** : numéro de la facture, type entier.
- **NCommande** : numéro de la commande, type entier.
- **date** : la date de la commande, type Date.
- **MontantHT** : le montant des articles de la commande sans taxe, type réel.
- **MontantTTC** : le montant des articles de la commande avec taxe (19%), type réel.
- **ListeArticle** : pointeur sur la Pile des articles de la commande, type Aliste.
- **Suiv** : pointeur qui pointe à l'élément suivant pour faire le chainage, type Commande.

Les fonctions du type Commande :

- **EmpilerCom** : cette procédure marche comme empiler dans une pile tout simplement, on la donne dans les paramètres la Pile des Commandes et la commande à empiler, et elle va l'empiler dans la Pile.
- **AjouCom** : cette procédure reçoit en paramètres la Pile des commandes, pile des articles et la file des personnes, alloue un espace du type commande, après elle demande aux utilisateurs d'introduire les données du nouvelle commande à ajouter qui seront scanner directement dans les attributs du nouveau élément allouer, et après elle empile cette nouvelle commande.
- **DepilerCommande** : une fonction de type de retourne commande, qui marche comme dépiler où elle va dépiler le sommet de la pile et le renvoi dans le retourne.
- **SupprimerCommande** : une procédure qui reçoit en paramètres la Pile des commandes, demande de l'utilisateur d'introduire le numéro de la commande qu'il veut la supprimer, elle fait la recherche de cette commande en dépiler, test, empiler dans une pile temporaire jusqu'à que la commande sera trouvé, si la commande est trouvé alors elle ne sera pas empiler dans pile temporaire, et la boucle de recherche arrête, la boucle de reconstruction commence où on dépile de la pile temporaire et on empile dans la Pile initial.
- **AffComList** : une procédure qui reçoit en paramètres la Pile des commandes, dépile le sommet de cette pile, faire l'affichage des attributs, empile dans une pile temporaire, et après quand la pile est vide, elle fait la reconstruction où elle dépile de la pile temporaire et empile dans la pile initial.
- **Affichage_cmd** : une procédure qui reçoit en paramètres la pile des commandes et le numéro de la commande recherché, dépile le sommet de cette pile, tester si le numéro de la commande dépiler égal au numéro de la commande rechercher alors faire l'affichage sinon faire empiler dans la pile temporaire et reboucler jusqu'à qui trouve la commande ou la pile et vide, faire la reconstruction de la pile initial après.
- **Rech_par_client** : une procédure qui reçoit en paramètres la pile des commandes et le ID de la personne recherché, dépile le sommet de cette pile, tester si le ID du personne dépiler égal au ID du personne rechercher alors faire l'affichage de la commande sinon faire empiler dans la pile

temporaire et reboucler jusqu'à qui trouve la commande ou la pile et vide, faire la reconstruction de la pile initial après.

- **Rech_par_cmd** : une procédure qui reçoit en paramètres la pile des commandes et le numéro de la commande recherché, dépile le sommet de cette pile, tester si le numéro de la commande dépiler égal au numéro de la commande rechercher alors faire l'affichage du message que la commande existe sinon faire empiler dans la pile temporaire et reboucler jusqu'à qui trouve la commande ou la pile et vide, si la boucle sort avec la pile vide et il n'a pas trouvé la commande alors il affiche le message que la commande n'existe pas, enfin, faire la reconstruction de la pile initial après.
- **ModifierCommande** : une procédure qui reçoit en paramètres la Pile des commandes et la Pile des articles, demande de l'utilisateur d'introduire le numéro de la commande qu'il veut la faire un changement, cette procédure va lancer une recherche de cette commande, s'il existe alors un menu sera afficher contenant les changements possible dans les attributs de cette commande, et l'utilisateur doit choisir une option et faire le changement nécessaire, sinon si la commande n'existe pas alors un message sera afficher que la commande n'existe pas.

Déclaration du type Commande

typedef struct Commande Commande;

```
struct Commande
{
    int IDclient;
    int Type;    // si 0 alors Achat sinon si 1 alors Vente
    int NFacture;
    int NCommande;
    Date date;
    float MontantHT;
    float MontantTTC;
    Aliste *ListeArticle;
    Commande *suiv;
};
```


Déclaration du Pile des Commandes

```
typedef struct ComListe ComListe;
```

```
struct ComListe
```

```
{
```

```
    Commande *SommetC;
```

```
};
```

Statistique

-l'article le plus vendu : procédure « ArtPlusVendu » reçoit en paramètres la pile des articles, dépile le sommet dans une variable temporaire, dépile le nouveau sommet et fait le test entre les deux attribut « NBvente » de ces deux dernier, si le NBvente du sommet est plus grand que lui du variable temporaire alors permuter entre les deux , après empiler dans une pile temporaire pour ne pas perdre les élément de la pile initial et faire la reconstruction après faire le test avec tous les élément de la pile.

-nombre de commande par client : procédure « NBComClient » reçoit en paramètres la file des personnes, demande à l'utilisateur d'introduire un ID d'une personne, fait la recherche de cette personne, si la personne est trouvé alors il affiche « nombre de commande de cette personne est : ... », Et bien sûr elle fait la reconstruction après.

-pires client : procédure « PireClient » reçoit en paramètres la file des personnes, fait une boucle pour trouver la première personne du type client pour le sauvegarder dans une variable temporaire et lancer la boucle de comparaison où si on trouve un **client** avec moins de commande on le permuter avec la dans la variable temporaire, après on fait la reconstruction de la file, et après on fait un affichage avec la condition que le nombre de commande sera égal de la variable temporaire.

Si la file personne est vide on affiche qu'elle est vide ou si elle contient que des fournisseurs ou divers on affiche qu'elle ne contient pas de client pour tester.

L'Exécution du Programme

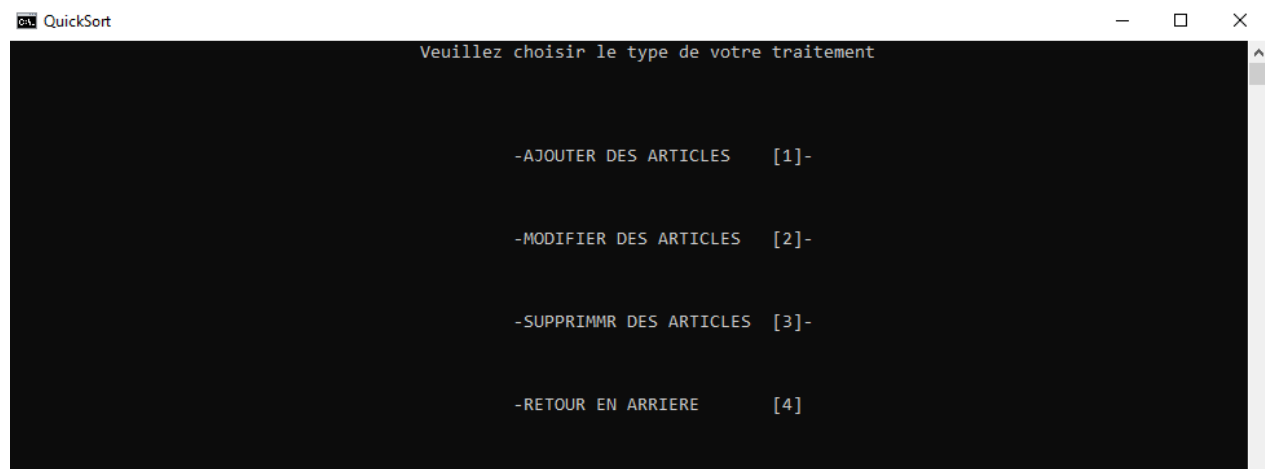
Menu d'accueil :



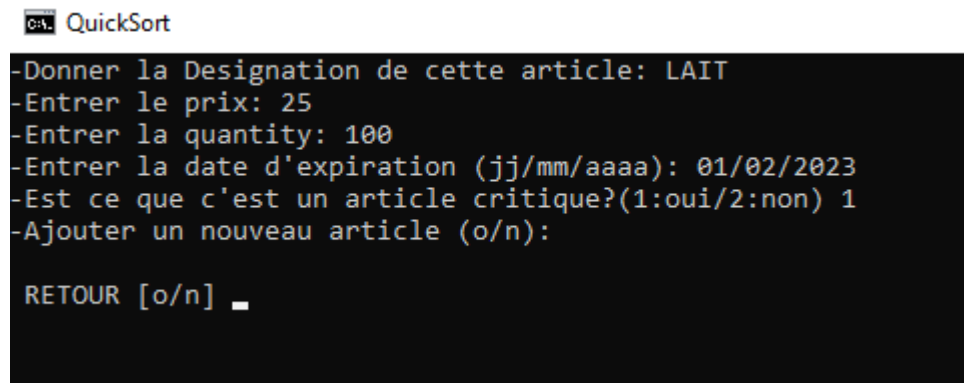
Traitement :



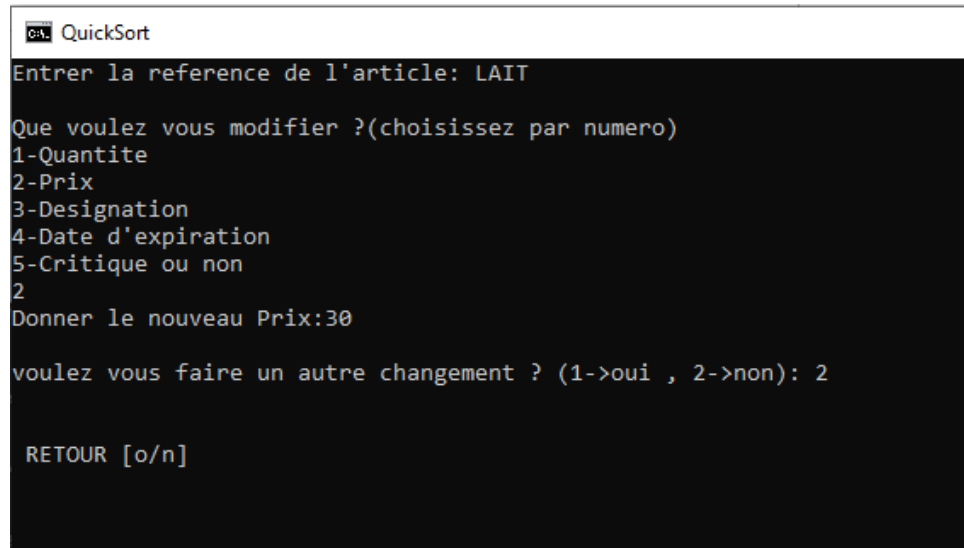
Gestion des articles :



Ajouter articles :



Modification articles :



Afficher tous les articles :

```
QuickSort
-Designation d'article :Iphone X
-Quantite : 10
-Prix de l'article: 500.00
-Date de peremption: 0/0/0
-Cet article n'est pas critique
-Designation d'article :LAIT
-Quantite : 100
-Prix de l'article: 30.00
-Date de peremption: 1/2/2023
-Cet article est critique
```

Supprimer des articles :

```
QuickSort
Donner la reference de l'article a supprimer: LAIT

RETOUR [o/n]
```

Gestion des commandes :

```
QuickSort
Veuillez choisir le type de votre traitement

-AJOUTER DES COMMANDES [1]-

-MODIFIER DES COMMANDES [2]-

-SUPPRIMER DES COMMANDES [3]-

-RETOUR EN ARRIERE [4]
```

Ajouter des commandes :

```
QuickSort
-Entrer l'ID du client: 112233
-Entrer le numero de la commande: 1234
-Entrer le Numero de la facture: 4321
-Entrer la date: 16/01/2023
-Donner le type de cette commande: (0->achat/1->vente) 1
donner l'article que vous voulez ajouter a la commande: LAIT
combien vous voulez:1
voulez-vous ajouter un autre article?(1-oui / 2-non)2
-Ajouter une nouvelle commande (o/n):

RETOUR [o/n]
```

Modifier des Commandes :

```
QuickSort
Donner le numero de commande que vous voulez changer: 1234

Que voulez vous modifier?(choisissez par numero)
1-Numero de la commande
2-Liste des Articles
3-Numero de la facture
4-Date
5-MontantHT
6-Type
3
Donner le nouveau numero de facture:12345

voulez vous faire un autre changement?(Si oui tapez 1)0

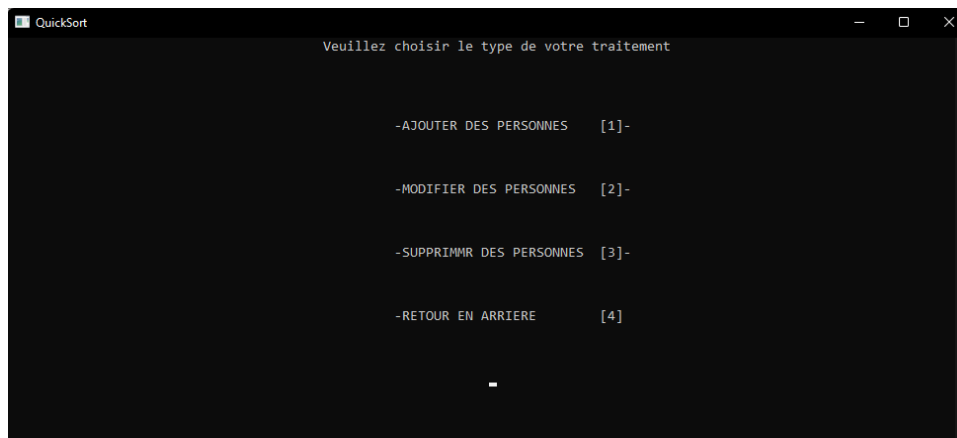
RETOUR [o/n]
```

Supprimer des commandes :

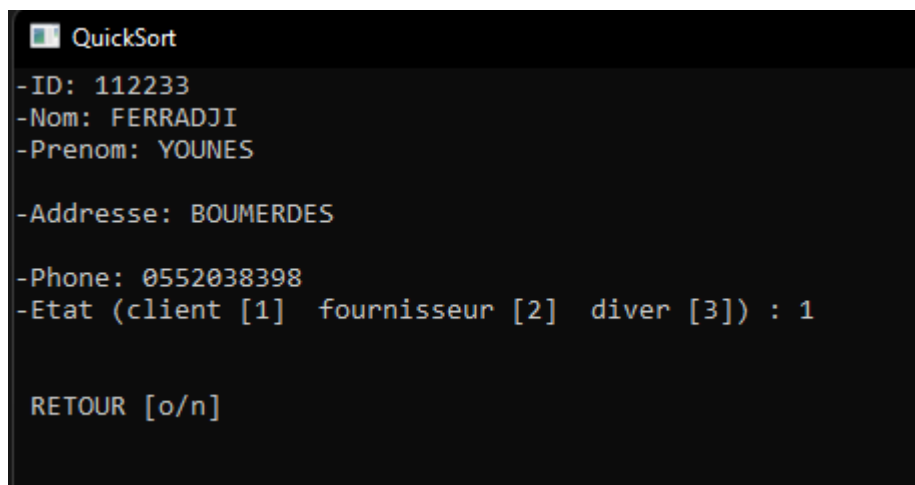
```
QuickSort
Donner le numero de la commande a supprimer: 1234

RETOUR [o/n]
```

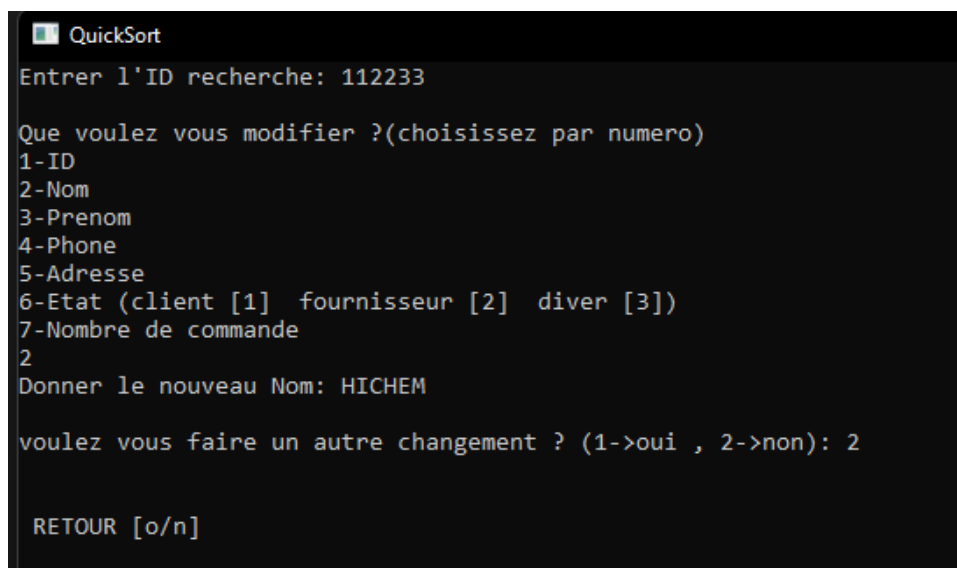
Gestion des personnes :



Ajouter des personnes :



Modifier des personnes :



Supprimer des personnes :

```
QuickSort

Donner ID de la personne qui vous voulez supprimer: 112233

RETOUR [o/n]
```

Recherche :

```
QuickSort

Selectionner un type de recherche

-RECHERCHER UN ARTICLE PAR REFERENCE [1]-

-RECHERCHER UN ARTICLE PAR DATE DE PEREMPTION [2]-

-RECHERCHER UNE COMMANDE PAR NUMERO DE COMMANDE [3]-

-RECHERCHER UNE COMMANDE PAR NUMERO CLIENT [4]-

-RETOUR AU MENU [5]-

-QUITTER [X]-
```

Rechercher un article par référence :

```
QuickSort

Entrer la reference de l'article: TRAMADOL
-Designation d'article :TRAMADOL
-Quantite dans le stock: 5
-Prix de l'article: 1000.00
-Date de peremption: 12/10/2023
-Cet article est critique

RETOUR [o/n]
```

Rechercher un article par date de péremption :

```
QuickSort
Entrer la date (jj/mm/aaaa): 12/10/2023
-Designation d'article :TRAMADOL
-Quantite dans le stock: 5
-Prix de l'article: 1000.00
-Date de peremption: 12/10/2023
-Cet article est critique

RETOUR [o/n] _
```

Recherche une commande par numéro de commande :

```
QuickSort
Entrer le Numero de commande :1234
Cette Commande figure dans la liste des commandes

RETOUR [o/n]
```

Recherche une commande par numéro client :

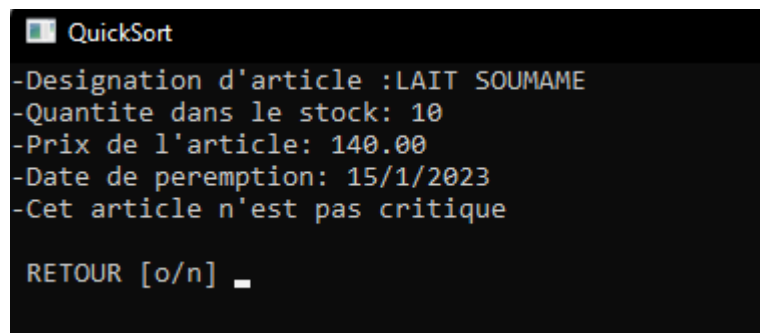
```
QuickSort
Entrer ID du client :112233
ID Client : 112233
N Facture : 12345
N Commande : 1234
Date : 16/1/2023
La liste des articles vendus : -Designation d'article :LAIT
-Quantite : 1
-Prix de l'article: 25.00
-Date de peremption: 16/2/2023
-Cet article n'est pas critique

RETOUR [o/n]
```

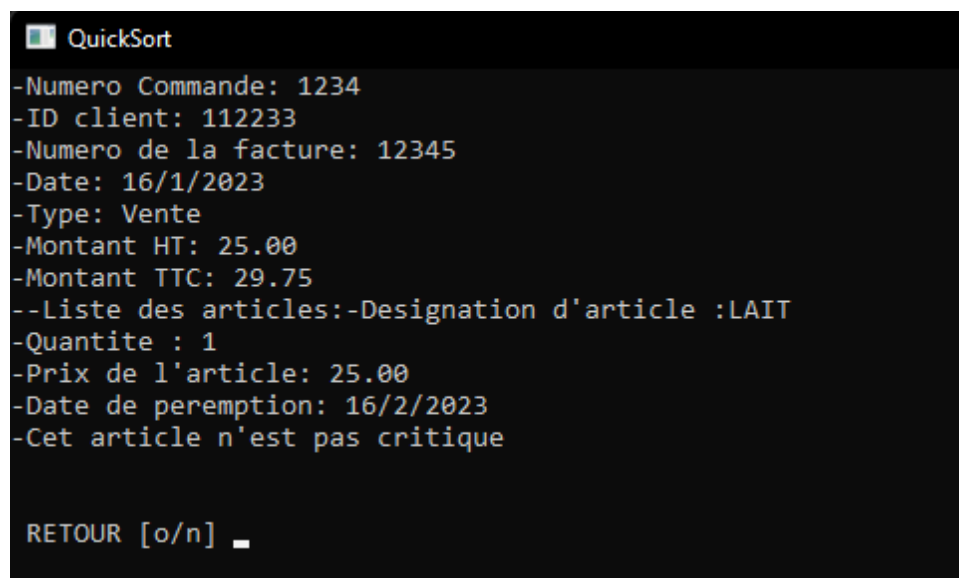

Affichage :



Afficher les produits expires :



Afficher tous les commandes :



Afficher une commande donnée :

```
QuickSort
Entrer le Numero de commande :1234
ID Client : 112233
N Facture : 12345
N Commande : 1234
Date : 16/1/2023
La liste des articles vendus : -Designation d'article :LAIT
-Quantite : 1
-Prix de l'article: 25.00
-Date de peremption: 16/2/2023
-Cet article n'est pas critique

RETOUR [o/n]
```

Affichage des consommables critiques :

```
QuickSort
-Designation d'article :TRAMADOL
-Quantite dans le stock: 5
-Prix de l'article: 1000.00
-Date de peremption: 12/10/2023
-Cet article est critique

RETOUR [o/n]
```

Statistique :

```
QuickSort
Selectionner quoi vous voulez afficher

-AFFICHER L'ARTICLE LE PLUS VENDU [1]-
-AFFICHER LE NOMBRE DE COMMANDE PAR CLIENT [2]-
-AFFICHER LE PIRE CLIENT [3]-
-RETOUR AU MENU [4]-

-QUITTER [X]-
```

Afficher l'article le plus vendu :

```
QuickSort
-L'article le plus vendu: LAIT

RETOUR [o/n]
```

Afficher le nombre de commande par client :

```
QuickSort
-Entrer l'ID de client: 112233
-Nombre des commande de FERRADJI est: 1

RETOUR [o/n] _
```

Afficher le pire client :

```
QuickSort

Le Pire Client est: SOMEONE WHO DOESNT CODE

RETOUR [o/n]
```