

TP 12 (pile, file)

Dans cet exercice, on souhaite simuler la gestion d'un stock de produits frais. Chaque produit est décrit par sa référence et sa date d'entrée en stock.

Soit le type *produit* et les procédures permettant de saisir ou d'afficher un produit :

```
typedef char chaine10[11] ;
typedef struct{
    chaine10 reference;
    int dateEntStock;
} produit;

void saisir(produit *adr_prod, int dateJ){
    // met dans *adr_prod un produit saisi au clavier à la date dateJ
    printf("référence ?\n");
    scanf("%s",adr_prod->reference);
    adr_prod->dateEntStock=dateJ;
}

void afficher(produit prod){
    // affiche à l'écran le produit prod
    printf("%s %d\n",prod.reference,prod.dateEntStock);
}
```

La date est la date julienne, c'est-à-dire le numéro du jour dans l'année. La date du jour est une variable déclarée dans le `main()`.

On souhaite proposer à l'utilisateur ce menu :

- 1) entrée d'un produit dans le stock,**
- 2) vente d'un produit sorti du stock,**
- 3) le temps passe,**
- 0) quitter.**

Le troisième choix (le temps passe) incrémente la date du jour, d'où la procédure :

```
void modifdate(int* adr_datej){
    //tourne la page de l'éphéméride
    (*adr_datej)++;
}
```

Partie 1 : la pile

Le temps évoluant, les produits en stock sont plus ou moins frais et l'on se place dans le cas où la gestion du stock s'effectue selon la doctrine : "on sort du stock le produit le plus frais (le plus récemment entré en stock)" : on utilisera donc une **pile**.

Question 1

Proposer et tester les primitives pour manipuler une structure de données de type **pile**.

Question 2

Ecrire un programme, utilisant les procédures spécifiées ci-dessous (init, entree, sortie), permettant de simuler la gestion du stock, sachant que :

- quand il veut entrer un produit dans le stock, l'utilisateur saisit au clavier la référence du produit, et la date d'entrée en stock est la date du jour (donc utilisation de la procédure saisir).
- à la sortie d'un produit du stock, le produit ne peut être vendu si sa date d'entrée dans le stock est supérieure de 3 jours à celle de la date du jour. Si le produit le plus frais ne peut être vendu, on sortira tous les produits, qui ne peuvent être vendus, avec un message *ad hoc*.

On utilisera le type `stock`, défini comme suit : `typedef pile stock ;`

et les entêtes des procédures :

`void init(stock* adr_st)` qui initialise le stock `*sadr_t` à vide,

`void entree(produit pr, stock* adr_st)` qui entre le produit `pr` dans le stock `*adr_st`,

`void sortie(stock* adr_st, int dateJ)` qui sort du stock `*adr_st` un ou des produit(s) connaissant la date du jour `dateJ`.

Ces procédures font appel aux primitives vues en cours concernant les piles et mises au point à la question 1.

Partie 2 : la file

Le temps évoluant, les produits en stock sont plus ou moins frais et l'on se place dans le cas où la gestion du stock s'effectue selon la doctrine : "on sort du stock le produit le moins frais (le plus anciennement entré en stock)" : on utilisera donc une **file**.

Question 1

Proposer et tester les primitives pour manipuler une structure de données de type **file**.

Question 2

Le programme principal simulant la gestion du stock ne change pas par rapport au programme précédent.

Après avoir redéfini le type stock comme suit :

```
typedef file stock;
```

modifier le contenu des procédures init, entree, et sortie (mêmes entêtes et mêmes spécifications que pour la stratégie « pile ») :

- void init(stock* adr_st) qui initialise le stock *adr_st à vide,
- void entree(produit pr, stock* adr_st) qui entre le produit *pr* dans le stock *adr_st,
- void sortie(stock* adr_st, int dateJ) qui sort du stock *adr_st un ou des produit(s) connaissant la date du jour *dateJ*.

Ces procédures font appel aux primitives vues en cours concernant les **files** et mises au point à la question 1.

N.B. Pour tenir compte de la contrainte liée à la date de péremption, on sortira tous les produits ne pouvant être vendus, en affichant les produits sortis du stock avec un message *ad hoc*, puis le produit vendu, ou le message "aucun produit à vendre".