

Série N° 1

Exercice 1 :

On considère le type **point2D** pour représenter les points du plan. Chaque point est caractérisé par deux champs x et y, qui sont les coordonnées d'un point dans le plan.

1. Donner le type de données **point2D** ;
2. Ecrire la fonction **point2D creerPoint (float x , float y)** qui permet de créer un point du plan ;
3. Ecrire une fonction qui affiche les coordonnées d'un point.
4. On veut saisir les points à l'aide du clavier, écrire la fonction **point2D saisie()** qui renvoie le point saisi.
5. Écrire un type **vect2D** pour représenter un vecteur dans le plan.
6. Ecrire les fonctions qui calculent la norme d'un vecteur, la distance entre deux points et le produit scalaire entre deux vecteurs.
7. Ecrire la fonction **main()** test.

Exercice 2 :

Un produit est caractérisé par son numéro un entier **num**, par son **nom** une chaîne de 10 caractères, par son **prix** unitaire un réel et par sa **date d'expiration (de)** définie par trois entier jour mois et année.

1. Donnez le type de données **Produit** adapté à la représentation des informations apparaissant dans l'énoncé ci-dessus ;
2. Ecrire la fonction **saisie()** permettant de saisir les informations relatives à un produit;
3. Ecrire la fonction **affiche (Produit p)** permettant d'afficher les informations relatives à un produit p donné ;
4. Soit **tab** un tableau de produit de taille TMAX=100, écrire les fonctions **saisieTab(Produit tab[], int n)** et **afficheTab(Produit tab[], int n)** pour la saisie et l'affichage de n éléments du tableau.
5. Ecrire la fonction **somPrix(Produit tab[], int n)** qui calcule la somme des prix des produits ;
6. Ecrire la fonction **afficheDE(Produit tab[], int n, Date dexp)** qui affiche le produit dont la date d'expiration est inférieur à la date **dexp**.
7. On veut rechercher un produit par son nom, écrire la fonction **rechSeq(Produit tab[], int n, char *nomP)** qui renvoie son indice i si le produit recherché existe dans tab et -1 sinon.

8. On suppose maintenant que le tableau `tab` est trié par ordre alphabétique des noms des produits, écrire la **fonction récursive** de recherche dichotomique qui renvoie son indice `m` si le produit recherché existe dans `tab` et `-1` sinon suivante :

```
rechDicho(Produit tab[], char * nomP, int i, int j)
```
9. Ecrire la fonction `main()` de test, afin qu'on n'aura pas un gaspillage de mémoire faire allouer les `n` éléments du tableau d'une manière dynamique.

Exercice 3 :

Une entreprise vend des fournitures de bureau (crayon, gomme, ciseaux,...). Chaque fourniture stockée dans la réserve est identifiée par un code numérique `id` (ex : `id=345`), un libellé **lib** exprimé sous la forme d'une chaîne de 10 caractères (ex. stylo feutre), un prix d'achat unitaire **prix** au fournisseur (ex. 10,5 DH), le nombre d'items achetés au fournisseur **nach** et le nombre d'items déjà vendus **nvend**.

Chaque fin d'année, l'entreprise effectue son inventaire, i.e. décompte pour chaque fourniture, le nombre d'items en stock (c.à.d. le nombre d'items achetés au fournisseur moins le nombre d'items vendus) puis calcule le montant total du stock (somme des prix d'achat unitaire multipliés par les nombres d'items en stock).

On vous demande d'écrire les structures et les fonctions suivantes :

1. Un type de données **Fourniture** adapté à la représentation des informations apparaissant dans l'énoncé ci-dessus ;
2. La fonction permettant de saisir les informations relatives à une fourniture et de les transmettre en résultat ;
3. La fonction permettant d'afficher les informations relatives à une fourniture donnée ;
4. La fonction qui met à jour une fourniture **f** donnée, quand une vente de **n** unités de cette fourniture vient d'être effectuée ;
5. La fonction permettant de calculer et de transmettre le nombre d'items en stock et le montant correspondant pour une fourniture **f** donnée ;
6. Un type de données **Catalogue** permettant de représenter l'ensemble des fournitures (au plus 500) vendues par l'entreprise.
7. La fonction permettant de rajouter une nouvelle fourniture **f** donnée dans un catalogue **c** donné.
8. La fonction bilan qui étant donné un catalogue **c**, affiche pour chaque fourniture, son libellé, le nombre d'items en stock et le montant correspondant, puis enfin, affiche la valeur totale du stock;
9. La fonction permettant de retirer une fourniture de code donné d'un catalogue **c** donné.