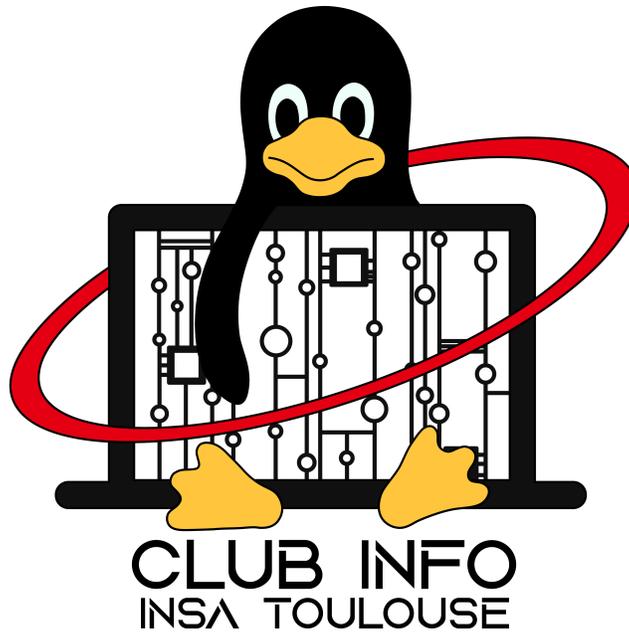


# Gestion d'une application de supermarché

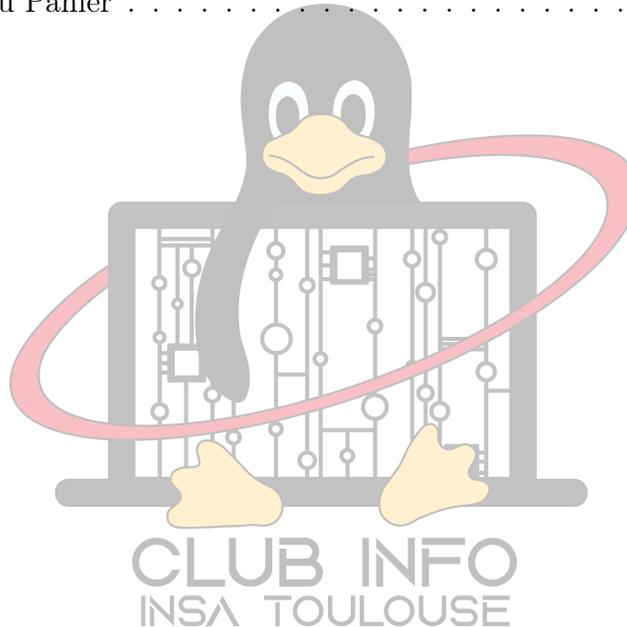
Paul Slisse

Novembre 2023



## Table des matières

<b>1</b>	<b>Utilisation d'un Tableau Dynamique</b>	<b>3</b>
1.1	Structure de Données : Produit et Catégories . . . . .	3
1.2	Dimensionnement Dynamique du Tableau : Création d'un Tableau de Produits . . . . .	3
1.3	Calcul du Total du stock . . . . .	3
1.4	Recherche de Produit avec Recommandation . . . . .	3
1.5	Sélection de Produit : Trouver le Produit le Plus Cher ou le Moins Cher	3
1.6	Algorithme de Tri : Tri par Prix Croissant . . . . .	3
<b>2</b>	<b>Utilisation d'une Liste Chaînée</b>	<b>4</b>
2.1	Définition d'une Liste de Produits et Fonction d'Ajout au Panier . . . .	4
2.2	Fonction de remplissage d'un panier . . . . .	4
2.3	Algorithmes Récursifs avec Liste Chaînée : Calcul du Total du Panier .	4
2.4	Algorithme de Recherche dans une Liste Chaînée : Recherche par Code dans le Panier . . . . .	4
2.5	Gestion des Promotions : Appliquer une Réduction de Prix dans le Panier	4
2.6	Affichage du Panier . . . . .	4
<b>3</b>	<b>Correction</b>	<b>5</b>



# 1 Utilisation d'un Tableau Dynamique

Dans cette première partie, votre supermarché préféré souhaite lier son stock avec son application pour pouvoir afficher les disponibilités en temps réel.

## 1.1 Structure de Données : Produit et Catégories

Concevez un type de record appelé Produit avec les champs suivants : Code (Entier), Nom (String), Prix (Float), Quantité (Entier) et Promotion (Boolean).

## 1.2 Dimensionnement Dynamique du Tableau : Création d'un Tableau de Produits

Définissez un tableau de produits. Écrivez une fonction qui demande au salarié de saisir le nombre de produits qu'il a en stock. Utilisez cette information pour dimensionner dynamiquement le tableau de produits. Ensuite, remplissez le tableau avec des produits (toujours en les demandant à l'utilisateur et toujours dans la fonction). On considérera que les produits avec un code multiple de 5 sont en promotion et les autres non (on gardera cette convention dans tout le reste de l'exercice). N'oubliez pas de tester la fonction dans le programme principal.

## 1.3 Calcul du Total du stock

Votre supermarché préféré souhaite connaître la valeur de son stock. Écrivez une fonction récursive qui prend le tableau de produits et renvoie le montant total du stock (on ne prendra en compte les promotions ici). N'oubliez pas de tester la fonction dans le programme principal.

## 1.4 Recherche de Produit avec Recommandation

Votre application de gestion de supermarché souhaite inclure une fonctionnalité de recommandation. Écrivez une fonction de recherche qui prend en entrée un code de produit et renvoie le produit correspondant dans le stock. Si le produit est introuvable, la fonction doit renvoyer une recommandation pour un produit populaire (Celui que vous voulez mais qui est dans le stock). N'oubliez pas de tester la fonction dans le programme principal.

## 1.5 Sélection de Produit : Trouver le Produit le Plus Cher ou le Moins Cher

Imaginez que vous travaillez sur une fonctionnalité de comparaison des prix dans votre application de gestion de supermarché. Écrivez une fonction qui, selon le choix de l'utilisateur, renvoie soit le produit le moins cher, soit le produit le plus cher dans le stock de produits. N'oubliez pas de tester la fonction dans le programme principal.

## 1.6 Algorithme de Tri : Tri par Prix Croissant

Votre supermarché souhaite afficher les produits sur son site Web dans l'ordre croissant des prix. Implémentez un algorithme de tri pour trier le stock de produits par prix croissant. N'oubliez pas de tester la fonction dans le programme principal.

## 2 Utilisation d'une Liste Chaînée

Dans cette partie, votre supermarché fait appel à vous pour mettre en place un nouveau concept, le drive. Les gens peuvent remplir leur panier en ligne, le payer puis venir le chercher en magasin sans avoir à passer du temps à l'intérieur du magasin (merci la société de consommation d'avoir inventé ce concept).

### 2.1 Définition d'une Liste de Produits et Fonction d'Ajout au Panier

Définissez une liste de produits qui représentera le panier d'un client. De plus, écrivez une fonction permettant d'ajouter un produit dans le panier d'un consommateur, on demandera toutes les infos du produit dans la fonction. N'oubliez pas de tester la fonction d'ajout dans le programme principal en créant le panier d'un consommateur.

### 2.2 Fonction de remplissage d'un panier

On va maintenant implémenter la fonctionnalité de remplissage d'un panier. Ecrivez une fonction qui parcourt le stock du supermarché (partie 1) et qui pour chaque produit demande au client quelle quantité il en veut (vérifier qu'il y en a assez dans le stock) et ajoute la quantité correspondante au panier.

### 2.3 Algorithmes Récursifs avec Liste Chaînée : Calcul du Total du Panier

Pour pouvoir finaliser son concept de drive, le supermarché veut que les clients puissent payer en ligne. Adaptez la fonction récursive de la partie 1 pour calculer le montant total du panier d'un consommateur en utilisant une liste chaînée (on ne prend toujours pas en compte les promotions). N'oubliez pas de tester la fonction dans le programme principal.

### 2.4 Algorithme de Recherche dans une Liste Chaînée : Recherche par Code dans le Panier

Le supermarché veut savoir si un produit appartient au panier d'un certain client. Adaptez la fonction de recherche pour travailler avec une liste chaînée représentant le panier d'un consommateur, on cherchera avec le code de l'article. N'oubliez pas de tester la fonction dans le programme principal.

### 2.5 Gestion des Promotions : Appliquer une Réduction de Prix dans le Panier

Le supermarché propose régulièrement des promotions et veut que son programme applique directement les promotions en ligne. Écrivez une fonction pour appliquer une réduction de prix de 10% à tous les produits en promotion dans le panier, puis n'oubliez pas de dire que le produit n'est plus en promotion pour ne pas ré-appliquer la promotion. N'oubliez pas de tester la fonction dans le programme principal en recalculant le prix après promotion.

### 2.6 Affichage du Panier

Le client aimerait bien voir son panier avant de le payer, écrivez une procédure permettant au client de visualiser son panier.

NB : N'oubliez pas de libérer l'espace mémoire pour éviter de potentiels problèmes.

### 3 Correction

```

1 with Ada.Text_IO;use Ada.Text_IO;
2
3 procedure Supermarche is
4 -- Partie 1: Utilisation d'un Tableau Dynamique
5
6 --Question 1
7 type CategorieProduit is (Epicerie, ProduitsFrais, Hygiene);
8
9 type Produit is record
10     Code : Integer;
11     Nom: String(1..20);
12     Prix : Float;
13     QuantiteEnStock : Integer;
14     --Categorie : CategorieProduit;
15     Promotion : Boolean;
16 end record;
17
18 --Question 2
19 type Tab_Produits is array(Integer range<>) of Produit;
20
21 function CreerTableauProduits return Tab_Produits is
22
23     Nb_Stock:Integer:=0;
24     Nom_Prod:String(1..20);
25     Code_Prod:Integer;
26     Prix_Prod:Float;
27     Quantite_Prod:Integer;
28     --Cat_Prod:CategorieProduit;
29     Promotion:Boolean;
30
31 begin
32     Put_Line("Combien de produits differents avez vous en stock?");
33     Get(Nb_Stock);
34 declare
35     Stock : Tab_Produits(1..Nb_Stock);
36 begin
37     for K in 1..Nb_Stock loop
38         Put_Line("Quel est le code du produit?");
39         Get(Stock(K).Code);
40         Put_Line("Quel est son nom?");
41         Get(Stock(K).Nom);
42         Put_Line("Quel est son prix?");
43         Get(Stock(K).Prix);
44         Put_Line("Quel quantite de se produit avez vous en stock?");
45         Get(Stock(K).Quantite);
46         if (Stock(K).Code mod 5)=0 then
47             Stock(K).Promotion:=True;
48         else
49             Stock(K).Promotion:=False;
50         end if;
51     end loop;
52     return Stock;
53 end CreerTableauProduits;
54
55 -- Question 3
56 function CalculerTotalStock(Stock : Tab_Produits) return Float is
57
58     Total : Float := 0.0;
59

```

```

60
61 begin
62     if Stock'First=Stock'Last then
63         Total:=Stock(Stock'First).Prix*Stock(Stock'First).Quantite;
64     else
65         Total:=CalculerTotalStock(Stock'First..Stock'First)+
66             CalculerTotalStock(Stock'First+1..Stock'Last);
67     return Total;
68
69
70 end CalculerTotalStock;
71
72 --Question 4
73 function RechercherProduitAvecRecommandation(Code : Integer; Stock :
74     Tab_Produit) return Produit is
75     ProduitRecherche : Produit;
76
77 begin
78     if Stock'Length=0 then
79         Put_Line("Le stock est vide");
80     else
81         for Produit in Stock loop
82             if Produit.Code = Code then
83                 ProduitRecherche:=Produit;
84             end if;
85         end loop;
86         -- Si le produit n'est pas trouve, recommander un produit populaire
87         ProduitRecherche := Stock(Stock'Length/2);
88     end if;
89     return ProduitRecherche;
90
91 end RechercherProduitAvecRecommandation;
92
93
94 --Question 5.
95 function ProduitLeMoinsOuLePlusCher(MoinsCher : Boolean; Stock : Tab_Produits)
96     return Produit is
97     ProduitSelectionne : Produit; --Va servir pour retenir le produit le moins/
98     plus cher
99
100 begin
101     if Stock'Length = 0 then
102         Put_Line("Le stock est vide.");
103     else
104         -- Initialiser avec le premier produit du panier
105         ProduitSelectionne := Stock(Stock'First);
106
107         -- Comparer les prix avec les autres produits dans le panier
108         for ProduitCourant in Stock(Stock'First+1..Stock'Last) loop
109             if MoinsCher and then ProduitCourant.Prix < ProduitSelectionne.Prix
110                 then
111                 ProduitSelectionne := ProduitCourant;
112             elsif not MoinsCher and then ProduitCourant.Prix > ProduitSelectionne
113                 .Prix then
114                 ProduitSelectionne := ProduitCourant;
115             end if;
116         end loop;
117     end if;

```

```

116     return ProduitSelectionne;
117
118 end ProduitLeMoinsOuLePlusCher;
119
120 -- Question 6
121 procedure TriParPrixCroissant(Stock : in out Tab_Produits) is
122
123     IndexMinimum : Natural;
124     Aux : Produit;
125
126 begin
127
128     for i in Stock'First .. Stock'Last - 1 loop
129         IndexMinimum := i;
130
131         --On cherche parmi les lments restants, le moins cher
132         for j in i + 1 .. Stock'Last loop
133             if Panier(j).Prix < Panier(IndexMinimum).Prix then
134                 IndexMinimum := j;
135             end if;
136         end loop;
137
138         -- changez les lments pour les trier
139         TempProduit := Panier(i);
140         Panier(i) := Panier(IndexMinimum);
141         Panier(IndexMinimum) := TempProduit;
142     end loop;
143
144 end TriParPrixCroissant;
145
146 -- Partie 2: Utilisation d'une Liste Cha n e
147
148 -- Question 1
149
150     type Cellule;
151     type Liste_Produit is access Cellule;
152     type Cellule is record
153         Prod:Produit;
154         Suiv:Liste_Produit;
155     end record;
156
157 procedure AddPanier(Prod : in Produit; Panier : in out Liste_Produit) is
158
159 begin
160
161     Panier := new Cellule'(Prod,Panier);
162
163 end AddPanier;
164
165 --Question 2
166
167 procedure Remplissage(Stock:in out Tab_Produits ,Panier: in out Liste_Produit)
168     is
169
170     Achat: Boolean;
171     Nombre: Integer;
172     P:Produit;
173
174 begin
175     for K in Stock'First..Stock'Last loop
176         Put_Line("Voulez vous acheter le produit:" & Stock(K).Nom & ",
177                 il en reste:" & Integer'Image(Stock(K).Quantite)

```

```

176         Get(Achat);
177         if Achat then
178             Put_Line("Combien en voulez vous?");
179             Get(Nombre);
180             if Nombre>Stock(K).Quantite then
181                 Put_Line("Vous ne pouvez pas en acheter autant
182                     , il n'y en a pas assez dans le stock.");
183             else
184                 Stock(K).Quantite:=Stock(K).Quantite-Nombre;
185                 P:=Stock(K);
186                 P.Quantite:=Nombre;
187                 AddPanier(P,Panier);
188             end if;
189         end loop;
190 end Remplissage;
191
192 --Question 3
193
194 function CoutPanier(Panier: in Liste_Produit) return Float is
195
196     Total:Float:=0.0;
197
198 begin
199
200     if Panier=NULL then
201         NULL;
202     else
203         Total:=Panier.all.Prod.Prix+CoutPanier(Panier.all.Suiv);
204     end if;
205     return Total;
206
207 end CoutPanier;
208
209 --Question 4
210
211 function InPanier(Code: Integer ; Panier: in Liste_Produit) return Boolean is
212
213     Aux:Liste_Produit:=Panier;
214     Appartient:Boolean:=False;
215
216 begin
217
218     while not(Appartient) and Aux!=NULL loop
219         if Aux.all.Prod.Code=Code then
220             Appartient:=True;
221         end if;
222         Aux:=Aux.all.Suiv
223     end loop;
224     return Appartient;
225 end InPanier;
226
227 --Question 5
228
229 procedure PromoPanier(Panier: in out Liste_Produit) is
230
231     P:Liste_Produit:=Panier;
232
233 begin
234
235     while P!=NULL loop
236         if P.all.Prod.Promotion then
237             P.all.Prod.Prix:=P.all.Prod.Prix*0.9;

```

```
237         P.all.Prod.Promotion:=False;
238     end if;
239     end loop;
240 end PromoPanier;
241
242 --Question 6
243
244 procedure AffPanier(Panier: Liste_Produit) is
245
246 begin
247     if Panier=NULL then
248         New_Line;
249     else
250         Put_Line("Code:" & Integer'Image(Panier.all.Prod.Code) & ",
                Nom:" & Panier.all.Prod.Nom & ", Prix:" & Float'Image(
                Panier.all.Prod.Prix) & ", Quantite:" & Integer'Image(
                Panier.all.Prod.Quantite));
251         AffPanier(Panier.all.Suiv);
252 end AffPanier;
253
254 -- Programme Principal
255
256 Stock : Tab_Produits := CreerTableauProduits();
257 Prod:Produit;
258 Panier:Liste_Produit:=NULL;
259
260 begin
261
262     -- Testez chaque fonction de la Partie 1 avec le tableau 'Stock'
263
264
265     --Affichage du stock
266     for Produit in Stock loop
267         Put_Line("Code:" & Integer'Image(Produit.Code) & ", Nom:" & Produit.
                Nom & ", Prix:" & Float'Image(Produit.Prix) & ", Quantite:" &
                Integer'Image(Produit.Quantite));
268     New_Line;
269
270     --Question 3
271     Put_Line("Prix total du stock : " & Float'Image(CalculerTotalStock(Stock))
                );
272     New_Line;
273
274     --Question 4
275     Prod:=RechercherProduitAvecRecommandation(2,Stock);
276     Put_Line("Code:" & Integer'Image(Prod.Code) & ", Nom:" & Prod.Nom & ", Prix
                :" & Float'Image(Prod.Prix) & ", Quantite:" & Integer'Image(Prod.
                Quantite));
277     New_Line;
278
279     --Question 5
280     Prod:=ProduitLeMoinsOuLePlusCher(True,Stock);
281     Put_Line("Produit le moins cher:" Put_Line("Code:" & Integer'Image(Prod.
                Code) & ", Nom:" & Prod.Nom & ", Prix:" & Float'Image(Prod.Prix) & ",
                Quantite:" & Integer'Image(Prod.Quantite));
282     New_Line;
283
284     --Question 6
285     TriParPrixCroissant(Stock);
286     for Produit in Stock loop
```

```
288     Put_Line("Code:" & Integer'Image(Produit.Code) & ", Nom:" & Produit.
        Nom & ", Prix:" & Float'Image(Produit.Prix) & ", Quantite:" &
        Integer'Image(Produit.Quantite));
289     New_Line;
290
291     -- Testez chaque fonction de la Partie 2 avec la liste cha n e 'Panier'
292
293     --Question 2
294     Remplissage(Stock,Panier);
295     AffPanier(Panier);
296     New_Line;
297
298     --Question 3
299     Put_Line("Total du panier : " & Float'Image(CoutPanier(Panier)));
300     New_Line;
301
302     --Question 4
303     Put_Line("Le produit 3 est-il dans le panier?");
304     Put_Line("R ponse: " & Boolean'Image(InPanier(3,Panier)));
305     New_Line;
306
307     --Question 5
308     PromoPanier(Panier);
309     Put_Line("Total du panier apr s r duction : " & Float'Image(CoutPanier(
        Panier)));
310     New_Line;
311
312     Free(Panier);
313
314 end Supermarche;
```

Listing 1 – Correction de l'exercice

