

1. L'énoncé

On se propose de développer un programme informatique qui simule le comportement d'un **distributeur de boisson**.

Les boissons à distribuer sont :

- un café chaud (sans sucre ou sucré (de 1 à 5 doses)) à 0,20 euros
- un chocolat chaud (sans sucre ou sucré (de 1 à 5 doses)) à 0,30 euros
- une bouteille de coca-cola light, à 1,20 euros.

Le distributeur accepte les pièces de 10cts, 20cts, 50cts, 1euro et 2euros.

Le distributeur rend la monnaie.

L'utilisateur introduit des pièces avant de choisir sa commande. Il peut annuler et récupérer ces pièces avant de choisir sa boisson.

Une fois le produit commandé, il n'y a pas la possibilité d'annuler la commande.

Si le distributeur ne peut pas rendre la monnaie alors le distributeur propose à l'utilisateur d'annuler ou d'accepter la commande. S'il accepte alors le distributeur prépare la commande mais ne rendra pas de monnaie.

Si le produit commandé n'est pas disponible alors le distributeur annule automatiquement la commande et rend les pièces introduites.

Une fois la commande préparée, le distributeur envoie un signal sonore.

2. La correction

Algorithme principal de comportement du distributeur

Debut

```
Initialiser le stock et les ingrédients des boissons du distributeur;  
Initialiser le stock des pièces du monnayeur du distributeur;  
Créer et afficher l'IHM contenant le tableau des commandes de  
l'utilisateur et contenant les éléments de visualisation des états du  
distributeur;
```

```
Afficher l'IHM;
```

```
Le nombre de pièces introduites est à 0;  
La quantité de dose de sucre est à 0;
```

Pour chacune des actions que l'utilisateur peut réaliser **Faire**

```
// Action 1
```

```
Si l'utilisateur a mis une pièce de monnaie alors  
ajouter cette pièce aux pièces introduites;
```

```
Finsi
```

```
// Action 2
```

```
Si l'utilisateur a choisi une quantité de dose de sucre alors  
mettre à jour la dose de sucre courante;
```

```
Finsi
```

```
//Action 3
```

```
Si l'utilisateur a choisi une boisson alors
```

```
  Si la boisson est disponible alors
```

```
    calculer le montant à rendre en fonction du prix de la boisson  
    et de la somme des pièces introduites;
```

```
  Si la monnaie à rendre est négative alors
```

```
    signaler que l'utilisateur n'a pas introduit suffisamment  
    de pièce;
```

```
  Sinon
```

```
    Si le monnayeur ne peut pas rendre la monnaie alors
```

```
      signaler cela à l'utilisateur et demander lui  
      s'il veut continuer ou annuler;
```

```
    Sinon
```

```
      rendre la monnaie à rendre;
```

```
      préparer la commande en fonction de la dose  
      de sucre courante;
```

```
      envoyer un signal sonore;
```

```
      mettre à 0 la dose de sucre courante;
```

```
      mettre à 0 les pièces introduites;
```

```
    Finsi
```

```
  Finsi
```

```
  Sinon
```

```
    signaler à l'utilisateur que la boisson est indisponible;
```

```
  Finsi
```

```
Finsi
```

```
//Action 4
```

```
Si l'utilisateur ne veut pas continuer alors
```

```
  rendre les pièces introduites;
```

```
  mettre à 0 la dose de sucre courante;
```

```
  mettre à 0 les pièces introduites;
```

```

Finsi

//Action 5
Si l'utilisateur veut continuer alors
    rendre la monnaie à rendre;
    préparer la commande en fonction de la dose
    de sucre courante;
    envoyer un signal sonore;
    mettre à 0 la dose de sucre courante;
    mettre à 0 les pièces introduites;
Finsi

//Action 6
Si l'utilisateur a choisi d'annuler alors
    rendre les pièces introduites;
    mettre à 0 la dose de sucre courante;
    mettre à 0 les pièces introduites;
Finsi

FinPour

Fin

```

Déterminer la monnaie à rendre

```

Debut
    monnayeur_tmp = monnayeur;
    Ajouter à monnayeur_tmp les pièces introduites;
    m = montant à rendre;
    pieces à rendre = vide;
    Pour valPiece de {2, 1, 0.50, 0.20, 0.10 } faire
        Tantque non fini faire
            Si m < valPiece Alors fini = vrai;
            Sinon
                Si le monnayeur_tmp contient valPiece alors
                    ajouter valPiece aux pieces à rendre;
                    m = m - valPiece;
                    Enlever du monnayeur_tmp valPiece;
                Sinon
                    fini = vrai;
            Finsi
        Finsi
    Fintantque
Finpour
Si m=0 alors
    le monnayeur peut rendre la monnaie;
    monnayeur = monnayeur_tmp;
Sinon
    le monnayeur ne peut pas rendre la monnaie;
Finsi

Fin

```