

Décomposition de la monnaie [ss05] - Exercice

Karine Zampieri, Stéphane Rivière

Unisciel  algoprogram  UNIVERSITÉ HAUTE-ALSACE Version 16 mai 2018

Table des matières

1	Décomposition de la monnaie / pgmonnaie	2
1.1	Décomposition du problème	2
1.2	Procédures et Programme	3

alg - Décomposition de la monnaie (TD)



Mots-Clés Algorithmes paramétrés ■

Requis Structures de base, Structures conditionnelles ■

Difficulté ●○○ (30 min) ■



Objectif

Cet exercice reprend l'exercice @[Décomposition de la monnaie] en réalisant une décomposition en procédures et fonctions et traite le cas des coupures nulles.

(image : google/images)



1 Décomposition de la monnaie / pgmonnaie

1.1 Décomposition du problème

L'exercice @[Décomposition de la monnaie] décompose une somme d'argent (en euros) en son équivalent minimal en billets de 100 €, 50 €, 10 €, et de pièces de 2 € et 1 €.

((alg)) Algorithme initial

```

Algorithme pgmonnaie1
Variable somme : Entier
Variable b100 , r100 : Entier
Variable b50 , r50 : Entier
Variable b10 , r10 : Entier
Variable p2 , r2 : Entier
Variable p1 : Entier
Début
| Afficher ( "Somme à décomposer? " )
| Saisir ( somme )
| b100 <- DivEnt ( somme , 100 )
| r100 <- Modulo ( somme , 100 )
| Afficher ( "==> " , b100 : 4 , " billet(s) de 100 euros, reste " , r100 )
| b50 <- DivEnt ( r100 , 50 )
| r50 <- Modulo ( r100 , 50 )
| Afficher ( "==> " , b50 : 4 , " billet(s) de 50 euros, reste " , r50 )
| b10 <- DivEnt ( r50 , 10 )
| r10 <- Modulo ( r50 , 10 )
| Afficher ( "==> " , b10 : 4 , " billet(s) de 10 euros, reste " , r10 )
| p2 <- DivEnt ( r10 , 2 )
| r2 <- Modulo ( r10 , 2 )
| Afficher ( "==> " , p2 : 4 , " piece(s) de 2 euros, reste " , r2 )
| p1 <- r2
| Afficher ( "==> " , p1 : 4 , " piece(s) de 1 euro" )
Fin

```

Exemple d'exécution

(Avec cet algorithme)

```

Somme à décomposer? 1254
==> 12 billet(s) de 100 euros, reste 54
==> 1 billet(s) de 50 euros, reste 4
==> 0 billet(s) de 10 euros, reste 4
==> 2 piece(s) de 2 euros, reste 0
==> 0 piece(s) de 1 euro

```



Proposez une décomposition en procédures et fonctions en écrivant les profils de ces dernières.

Aide simple

Constatez que trois lignes se reproduisent.

1.2 Procédures et Programme



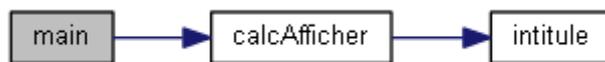
Écrivez une procédure `calcAfficher(somme, valeur, nc, rt, typec)` qui, pour une somme d'argent `somme` (entier) et une valeur de coupures `valeur` (entier), calcule le nombre de coupures dans `c` (entier), le reste à décomposer dans `rt` (entier) et affiche le résultat suivant (où `[x]` désigne le contenu de `x`) :

```
==> [nc] [typec] de [valeur] euros, reste [rt]
```

Le type de la coupure est défini par le caractère `typec` qui vaut 'b' (billet) ou 'p' (pièce). Dans le cas où `nc` est nul, n'affichez pas le texte.



Ré-écrivez un algorithme en utilisant la procédure `calcAfficher`.



Aide simple

Pour calculer le nombre de pièces de 1 euro, ajoutez une variable `r1` pour l'appel.