
Cours d'algorithmique

Arthur Blaise

oct. 04, 2018

Contenu:

1	Algorithmique et programmation procédurale	1
1.1	TD Éléments de base – Structures	1

Algorithmique et programmation procédurale

1.1 TD Éléments de base – Structures

1.1.1 Exercices sur les pièces (#13)

Consigne : Entrer des valeurs de pièces (0.1€, 0.2€, 0.5€, 1€, 2€). Contrôler la saisie et arrêter quand la valeur de la pièce est nulle. Puis, constituer des rouleaux de 10 pièces de même valeur et des sacs de 20 rouleaux. Afficher alors, pour chaque type de pièces, le nombre de rouleaux, de sacs et de pièces non emballées.

```
PROGRAMME tic-tac-boum
VARIABLES
    n1,n2,n5,n10,n20 : ENTIER //nombre de pièces par montant
    r1,r2,r5,r10,r20 : ENTIER //nombre de rouleaux
    s1,s2,s5,s10,s20 : ENTIER //nombre de sacs
DEBUT
    n2,n5,n10,n20 <- 0,0,0,0,0
    //on récupère le nombre de pièces de 0.1e
    ECRIRE("Nombre de pièces de 0.1€")
    LIRE(n1)
    SI (n1<>0) ALORS :
        //idem pour les pièce de 0.2e
        ECRIRE("Nombre de pièces de 0.2€")
        LIRE(n2)
    SI (n2<>0) ALORS :
        //idem pour 0.5e
        ECRIRE("Nombre de pièces de 0.5€")
        LIRE(n5)
    SI (n5<>0) ALORS :
        //idem pour 1e
        ECRIRE("Nombre de pièces de 1€")
        LIRE(n10)
    SI (n10<>0) :
        //idem pour 2e
```

(suite sur la page suivante)

(suite de la page précédente)

```

                ECRIRE("Nombre de pièces de 2€")
                LIRE(n20)
            FIN SI
        FIN SI
    FIN SI
FIN SI

//r1 récupère le nombre de rouleaux complets de 0.1 et n1 le reste,
// le tout grâce à une division euclidienne
r1,n1 <- n1 DIV 10, n1 MOD 10
//s1 récupère le nombre de sacs complets de 0.1e et r1 tout les rouleaux restants
s1,r1 <- n1 DIV 10, n1 MOD 20
//la suite des instructions est sur la même base, seuls les valeurs changent
r2,n2 <- n2 DIV 10, n2 MOD 10
s2,r2 <- r2 DIV 10, r2 MOD 20
r5,n5 <- n5 DIV 10, n5 MOD 10
r5,n5 <- r5 DIV 10, r5 MOD 20
r10,n10 <- n10 DIV 10, n10 MOD 10
s10,r10 <- r10 DIV 10, r10 MOD 20
r20,n20 <- n20 DIV 10, n20 MOD 10
s10,r20 <- r20 DIV 10, r20 MOD 20

//puis on affiche le résultat dans un texte sur plusieurs lignes
ECRIRE("Il y a ",s1," sacs ",r1," rouleaux ",p1," pièces de 0.1€, \n",
      s2," sacs ",r2," rouleaux ",p2," pièces de 0.2€, \n",
      s5," sacs ",r5," rouleaux ",p5," pièces de 0.5€, \n",
      s10," sacs ",r10," rouleaux ",p10," pièces de 1€, et \n",
      s20," sacs ",r20," rouleaux ",p20," pièces de 2€")
FIN

```

Plusieurs améliorations sont possible, comme par exemple créer une fonction qui retourne le nombre de rouleaux et de sacs possibles à partir du nombre de pièces, ou utiliser une boucle POUR afin de réduire la taille du premier bloc de code, comme ceci :

```

[... ]
DEBUT
    n2,n5,n10,n20 <- 0,0,0,0,0
    POUR i DANS [(n1,0.1),(n2,0.2),(n3,0.5),(n4,1),(n5,2)] FAIRE :
        ECRIRE("Nombre de pièces de ",i[1],"€")
        LIRE(i[0])
        SI (i[0]=0) ALORS:
            BREAK
[... ]

```

1.1.2 Exercice de l'horloge (#14)

Consigne : Créer un compte à rebours (heures, minutes, secondes) qui affichera «boum» à 00:00:00

```

PROGRAMME tic-tac-boum
VARIABLES
    h,m,s : ENTIER //heures, minutes et secondes restantes
DEBUT
    ECRIRE("Saisissez le temps restant (heures, minutes et secondes) : ")
    LIRE(h,m,s)
    TANT QUE (h<>0 OU m<>0 OU s<>0) :

```

(suite sur la page suivante)

(suite de la page précédente)

```
s <- s-1
SI (s<0 ET (m>0 OU h>0) ALORS :
  m <- m-1
  s <- 59
  SI (m<0 ET h>0) ALORS :
    h <- h-1
    m <- 59
  FIN SI
FIN SI
Ecrire(h, ":", m, ":", s)
FIN TANT QUE
Ecrire("BOUM !")
FIN
```

On pourra éventuellement ajouter avant la boucle TANT QUE une sécurité permettant d'empêcher un nombre supérieur à 59 de minutes ou de secondes :

```
TANT QUE (s>59) :
  s <- s-60
  m <- m+1
FIN TANT QUE
TANT QUE (m>59) :
  m <- m-60
  h <- h+1
FIN TANT QUE
```

Enfin, pour plus de réalisme, il est possible d'ajouter une instruction demandant au code d'attendre une seconde, dans la boucle TANT QUE. Cela permettra d'attendre une seconde entre deux décomptes, au lieu de tout afficher quasi-instantanément.