

# Structures de base [bs] - Sujets d'examens

Karine Zampieri, Stéphane Rivière

Unisciel  algoprogram  Version 14 mai 2018

## Table des matières

1	Les ardoises d'un toit / pardoises (4 points)	2
2	Force d'attraction / pattraction (3 points)	4
3	Troupeau vaches-moutons / ptroupeau (4 points)	5
4	Références générales	6

## alg - Sujets d'examens (Solution)

# 1 Les ardoises d’un toit / pardoises (4 points)



## Objectif

Calculer le nombre de paquets d’ardoises pour couvrir un toit constitué de deux rectangles identiques. Les ardoises sont vendues par paquet de 12.

Voici un exemple du résultat attendu :

```
Hauteur et largeur d’un versant du toit? 3.5 10
Et celles d’une ardoise? 0.20 0.5
==> Nombre de paquets : 60
```



(0.5 point) Définissez les constantes entières :

- `NRECTANGLES=2` (nombre de rectangles).
- `NARDPAQUET=12` (nombre d’ardoises par paquet).



(1 point) Écrivez un algorithme qui saisit :

- Les dimensions d’un versant de toit dans `httoit` et `lrtoit`.
- Celles d’une ardoise dans `htard` et `lrard`.

Toutes les variables sont réelles.

Affichez les invites :

```
Hauteur et largeur d’un versant du toit?
Et celles d’une ardoise?
```



(1 point) Pour un versant, calculez l’arrondi supérieur du nombre d’ardoises :

- En hauteur dans `nardht` (entier) :  $nardht = \lceil httoit/htard \rceil$
- En largeur dans `nardlr` (entier) :  $nardlr = \lceil lrtoit/lrard \rceil$

( $\lceil x \rceil$  désigne l’entier supérieur du réel  $x$ )

## Outil alg

L’opération  $\lceil x \rceil$  s’écrit `Arrondi(x)`.



(0.5 point) Dans l’entier `nardoises`, calculez le nombre total d’ardoises :

$$nardoises = nardht \times nardlr \times NRECTANGLES$$



(0.5 point) Dans l’entier `npaquets`, calculez le nombre de paquets d’ardoises.

**Attention**, ici aussi il n’y a pas de demi-paquet d’où l’arrondi supérieur :

$$npaquets = \lceil nardoises/NARDPAQUET \rceil$$



(0.5 point) Affichez (où [x] désigne le contenu de x) :

==> Nombre de paquets : [npaquets]



Testez.



Validez votre algorithme avec la solution.

**Solution alg** @[pgardoises1.alg]

```
Algorithme pgardoises1
Constante NRECTANGLES <- 2
Constante NARDOISES_PAQUET <- 12
Variable httoit , lrtoit , htard , lrard : Réel
Variable nardht , nardlr , nardoises , npaquets : Entier
Début
| Afficher ( "Hauteur et largeur d'un versant du toit? " )
| Saisir ( httoit , lrtoit )
| Afficher ( "Hauteur et largeur d'une ardoise? " )
| Saisir ( htard , lrard )
| nardht <- Arrondi ( httoit / htard )
| nardlr <- Arrondi ( lrtoit / lrard )
| nardoises <- nardht * nardlr * NRECTANGLES
| npaquets <- Arrondi ( nardoises / NARDOISES_PAQUET )
| Afficher ( "==> Nombre de paquets : " , npaquets )
Fin
```

## 2 Force d’attraction / pgattraction (3 points)



### Objectif

Déterminer la force d’attraction entre deux corps.

Voici un exemple du résultat attendu :

```
Masses des deux corps? 1.5 2
Distance? 10
Force d’attraction vaut 0.2943
```



(1 point) Écrivez un algorithme qui saisit :

- La masse de deux corps dans `m1` et `m2`.
- La distance qui les sépare dans `r`.

Toutes les variables sont des réels.

Affichez les invites :

```
Masses des deux corps?
Distance?
```



(0.5 point) Définissez la constante `GTERRE=9.81` (gravitation `g`).



(1 point) Dans le réel `f`, calculez la force d’attraction définie par :

$$F = g \cdot m_1 \cdot m_2 / r^2$$



(0.5 point) Affichez (où `[x]` désigne le contenu de `x`) :

```
Force d’attraction vaut [f]
```



Testez.



Validez votre algorithme avec la solution.

### 3 Troupeau vaches-moutons / pgtroupeau (4 points)



#### Objectif

Déterminer le prix d'un troupeau composé de vaches et de moutons.  
Voici un exemple du résultat attendu :

```
Nombre de vaches? 3
Nombre de moutons? 7
Prix des vaches est 750 euros
Celui des moutons est 840 euros
Total du troupeau est 1590 euros
Nombre d'animaux est 10
```



(1 point) Écrivez un algorithme qui saisit :

- Le nombre de vaches dans `nv` (entier).
- Celui des moutons dans `nm` (entier).

Affichez les invites :

```
Nombre de vaches?
Nombre de moutons?
```



(0.5 point) Définissez les **constantes réelles** :

- `PRIX_VACHE=250.0` (prix d'une vache) en €.
- `PRIX_MOUTON=120.0` (prix d'un mouton) en €.



(0.5 point) Calculez le prix :

- Des vaches dans `prv` (réel).
- Celui des moutons dans `prm` (réel).



(1 point) Affichez (où `[x]` désigne le contenu de `x`) :

```
Prix des vaches est [prv] euros
Celui des moutons est [prm] euros
```



(1 point) Calculez et affichez :

```
Total du troupeau est ... euros
Nombre d'animaux est ...
```



Testez.



Validez votre algorithme avec la solution.

**Solution alg** @[pgtroupeau1.alg]

```
Algorithme PGTroupeau1
Constante PRIX_VACHE <- 250.0
Constante PRIX_MOUTON <- 120.0
Variable nv , nm : Entier
Variable prv , prm : Réel
Début
| Afficher ( "Nombre de vaches? " )
| Saisir ( nv )
| Afficher ( "Nombre de moutons? " )
| Saisir ( nm )
| prv <- PRIX_VACHE * nv
| prm <- PRIX_MOUTON * nm
| Afficher ( "Prix des vaches est " , prv , " euros" )
| Afficher ( "Celui des moutons est " , prm , " euros" )
| Afficher ( "Total du troupeau est " , prv + prm , " euros" )
| Afficher ( "Nombre d’animaux est " , nv + nm )
Fin
```

## 4 Références générales

Comprend ■