

Dernier jour d'un mois/année [ss02]

Exercice résolu

Karine Zampieri, Stéphane Rivière

Unisciel  algoprog  Version 16 mai 2018

Table des matières

1	Dernier jour d'une date / pgsjours	2
1.1	Énoncé	2
1.2	Fonction dernierJour0	2
1.3	Procédure de test	2
1.4	Bissextilité en terme de divisibilité	3
1.5	Fonction divisible (test de divisibilité)	3
1.6	Fonction bissextile	4
1.7	Procédure de test	4
1.8	Fonction dernierJour	4
1.9	Procédure de test revisitée	5
2	Compléments	5
2.1	Validité d'une date	5

C++ - Dernier jour d'un mois/année (TP)



Mots-Clés Algorithmes paramétrés ■

Utilise Fonction, Procédure de test ■

Requis Structures de base, Structures conditionnelles ■

Difficulté ●●○ (1 h) ■



Objectif

Cet exercice calcule le dernier jour d'un mois et année donnés.

1 Dernier jour d'une date / pgsjours

1.1 Énoncé

Objectif

Déterminer le dernier jour d'un mois et année donnés.

Résultat attendu

Voici un exemple du résultat attendu :

```
De quel mois s'agit-il? 2
De quelle année? 2000
==> En 2000 le dernier jour du mois 2 est 29
```

1.2 Fonction dernierJour0

Commençons par déterminer le dernier jour d'un numéro de mois donné **indépendamment** d'une année.



Écrivez le **profil** d'une fonction `dernierJour0(mm)` qui renvoie le dernier jour d'un numéro de mois `mm`. Le mois est donné sous la forme d'un entier (1 pour janvier...).



Écrivez son corps.

1.3 Procédure de test



Écrivez une procédure `test_sjours0` qui saisit un numéro de mois (entier).
Affichez l'invite :

```
De quel mois s'agit-il?
```



Mémorisez le dernier jour du mois dans une variable (entier).



Affichez l'un des deux messages (où `[x]` désigne le contenu de `x`) :

```
==> Le dernier jour du mois [mois] est [njours]
==> Impossible ce mois n'existe pas
```



Écrivez un programme qui se contente d'appeler la procédure de test.



Testez. Exemples d'exécution :

```
De quel mois s'agit-il? 5
==> Le dernier jour du mois 5 est 31
```

```
De quel mois s'agit-il? 15
==> Impossible ce mois n'existe pas
```

1.4 Bissextilité en terme de divisibilité

Pour tenir compte du mouvement réel de la Terre, un Romain d'avant notre ère décréta qu'il y aurait des années bissextiles environ tous les quatre ans. Il fut décidé que les années de notre ère dont le numéro est multiple de 4 seraient bissextiles et que le jour supplémentaire serait le 29 février.

Mais cette correction est un peu trop forte, et un pape du nom de GRÉGOIRE décida que :



Propriété

Les années de siècles (c.-à-d. multiples de 100) ne sont pas bissextiles, sauf si elles sont multiples de 400.



Soient deux entiers n et d . Dire que d est multiple de n équivaut à dire que... (à vous de compléter) en terme de divisibilité.



Traduisez l'énoncé en terme de divisibilité : « [...] années multiples de 4 sont bissextiles. [...] mais que les années de siècles (c.-à-d. multiples de 100) ne sont pas bissextiles, sauf si elles sont multiples de 400. »

1.5 Fonction divisible (test de divisibilité)



Écrivez le **profil** d'une fonction `divisible(n,d)` qui renvoie `Vrai` si un entier n est divisible par un entier d , `Faux` sinon.



Propriété

Un entier n est **divisible** par un entier d si (et seulement si) le reste de la division entière de n par d (c.-à-d. le modulo) est nul.



Écrivez le corps de la fonction.

1.6 Fonction bissextile



Définition

Une année postérieure à 1592 (début du calendrier grégorien) est **bissextile** si elle est divisible par 4 mais **non** divisible par 100, ou si elle est divisible par 400.



Écrivez une fonction `bissextile(an)` qui teste et renvoie `Vrai` si le millésime d'une année `an` (entier), supposée postérieure à 1592, est bissextile, `Faux` sinon.



Aide simple

Écrivez une expression booléenne dépendant de `an` qui traduit l'énoncé.
Attention aux parenthèses de l'expression.

1.7 Procédure de test



Écrivez une procédure `test_bissextile` qui saisit le millésime d'une année (entier) puis affiche si elle est ou non bissextile.



Testez. Exemples d'exécution :

```
Annee? 2004
==> Vrai
```

```
Annee? 1900
==> Faux
```

```
Annee? 2000
==> Vrai
```

1.8 Fonction dernierJour

Ce problème revisite la fonction `dernierJour0` de sorte à tenir compte des années bissextiles.



Copiez/collez la fonction `dernierJour0` en la fonction `dernierJour(mm,an)` qui renvoie le dernier jour d'un numéro de mois `mm` (entier) et année `an` (entier) donnés.



Modifiez le cas de février (cas 2) :

- Si l'année est bissextile, mettez 29 dans le résultat `rs`, 28 sinon.

1.9 Procédure de test revisitée



Copiez/collez la procédure de test `test_sjours0` en la procédure `test_sjours` puis complétez-la de sorte qu'elle saisit également le millésime d'une année (entier).

Affichez l'invite :

```
De quelle année?
```



Modifiez le message en (où `[x]` désigne le contenu de `x`) :

```
En [annee] le dernier jour du mois [mois] est [njours]
```



Lancez-le avec les exemples d'exécution.

```
De quel mois s'agit-il? 5
De quelle année? 2010
==> En 2010 le dernier jour du mois 5 est 31
```

```
De quel mois s'agit-il? 15
De quelle année? 2010
==> Impossible ce mois n'existe pas
```

```
De quel mois s'agit-il? 2
De quelle année? 2000
==> En 2000 le dernier jour du mois 2 est 29
```

2 Compléments

2.1 Validité d'une date

Une date est mémorisée dans trois variables de type entier, une pour le numéro de jour, une pour le numéro du mois et une pour le millésime de l'année. Par exemple, la date du 12 février 2013 est composée de trois entiers (12, 2, 2013).



Définition

Une **date** (j, m, a) est **valide** si :

- L'année a est positive.
- Le mois m est compris entre 1 et 12.
- Le jour j est compris entre 1 et le nombre de jours du mois m de l'année a .



Écrivez une fonction `datum(jr,mm,an)` qui teste et renvoie `Vrai` si et seulement si le triplet d'entiers `(jr,mm,an)` représente une date, `Faux` sinon.



Écrivez une procédure `test_saisir` qui demande et saisit une date jusqu'à ce qu'elle soit valide.



Testez.