

TD 3: Prog.

5. À quelles valeurs s'évaluent les expressions suivantes selon les valeurs de x ?

1. $x \neq 0$ and $1/x > 0$
2. $1/x > 0$ and $x \neq 0$
3. $x == 0$ or $1/x > 10$

1. $x < 0$; false
 $x = 0$; false
 $x > 0$; true.

2. $x < 0$; false
 $x = 0$; zero error division
 $x > 0$; true.

3. $x = 0$; True.

$0 < x < 0.1$; True

$x < 0$ or $x > 0.1$; False.

6. Simplifier les expressions booléennes suivantes (c'est à dire donner des expressions qui s'évaluent au même résultat pour les mêmes valeurs de variables). On suppose que x , y et z sont associées à des valeurs de type booléen.

1. True or x
2. not x or False
3. x and False
4. True and y and not z
5. (not x) == True
6. x or not x

1. True

2. not x .

3. False

4. y and not z

5. $x == \text{false}$

6. True.

7. En utilisant une variable `meteo` pouvant être associée aux valeurs (de type `str`) "brouillard", "pluie" ou "soleil", exprimer en Python les assertions suivantes :

- a. il pleut, vrai ou faux ?
- b. soit il pleut, soit il fait soleil
- c. il est faux qu'il pleut
- d. il pleut et il y a du brouillard

a. `assert(meteo == "pluie")`

b. `assert(meteo == "pluie" or meteo == "soleil")`

c. `assert(meteo != "pluie")`

d. `assert(meteo == "pluie" and meteo == "brouillard")`

8. En utilisant une variable x , exprimer les assertions :

e. $\sqrt{x} > 3$ et x est positif

f. soit x est négatif, soit $\sqrt{x} > 3$

e. $x > 9$

f. $x < 0$ or $\text{sqrt}(x) > 3$

9. En utilisant une variable `digicode` qui peut être associée aux valeurs "accueil", "atelier" ainsi que les chaînes de longueur 1 ou 2 représentant un numéro de salle (par exemple "12", "5"), exprimer l'assertion :

g. on a sonné à l'atelier ou à une salle numérotée de 8 à 15 inclus.

(Utiliser l'opérateur de conversion `int`, ex : `int('1')` vaut 1).

g. `digicode == "atelier" or (digicode != "accueil" and  $8 \leq \text{int}(\text{digicode}) \leq 15$ )`

On s'intéresse à des propriétés sur les intervalles fermés définis dans les entiers.

10. Comment représenter un tel intervalle ? Quels seront les paramètres d'un prédicat dont la donnée d'entrée est un intervalle ? 2 intervalles ?

représenter intervalle : $a \leq x \leq b$

paramètres : `borne_inf: int; borne_sup: int`

2 intervalles : $a1 \leq x \leq b1$ or $a2 \leq x \leq b2$.

11. Étant donné un intervalle I défini par les variables `b_inf` et `b_sup` et une variable x , donner une expression Python exprimant le fait que x appartient à I .

$b_inf \leq x \leq b_sup$.

12. Étant donné deux intervalles $I1$ et $I2$ définis respectivement par les variables `b_inf1` et `b_sup1` d'une part et `b_inf2` et `b_sup2` d'autre part, donner une expression Python exprimant le fait que $I1$ et $I2$ sont d'intersection vide.

not ($b_inf \leq b_inf2 \leq b_sup$ or $b_inf \leq b_sup2 \leq b_sup$.)

13. Trouver un nom pour les prédicats suivants, et prévoir des tests :

1. Prédicat qui renvoie `True` ssi un nombre entier donné est dans un intervalle donné, bornes comprises

2. Prédicat qui renvoie `True` ssi l'intersection de 2 intervalles est vide

`element_est_dans_intervalle(x: int, b_inf: int, b_sup: int)`

\$\$\$ 3, 2, 5

True

\$\$\$ 0, -3, 3

True

\$\$\$ 7, 0, 0

False

\$\$\$ -5, 5, 7

False

intersection-est-vide ($b_inf1: int, b_sup1, b_inf2, b_sup2$)

\$\$\$ 1, 4, 7, 12

True

\$\$\$ 0, 4, 2, 7

False

\$\$\$ 0, 2, -3, -2

True

\$\$\$ 0; 0, 3; 3.

False.

5 Écriture de prédicats

L'année compte habituellement 365 jours. Les années bissextiles comptent 366 jours.

Depuis l'ajustement du calendrier grégorien (en 1582), l'année est bissextile :

- si l'année est divisible par 4 et non divisible par 100, ou
- si l'année est divisible par 400.

On veut écrire un prédicat qui prend en paramètre une **annee** et renvoie **True** ssi **annee** est bissextile.

14. Comment nommer ce prédicat ?

15. Avec la définition écrite plus haut, les années 2019, 2020, 2023 et 2000 sont-elles bissextiles ?

16. Comment calculer : l'entier x est divisible par l'entier y , vrai ou faux ?

17. Quelle décomposition du problème peut-on envisager ?

18. Écrire un prédicat qui renvoie vrai ssi son paramètre est une année bissextile.

14. `annee_est_bissextile`.

15. 2000, 2024 bissextile.

2019, 2023 ~~est~~ bissextile.

16. $x \% y == 0$.

17. Savoir si un nombre est divisible par un autre.

Savoir si l'année respecte les condit^s d'une année bissextile.

18. def annee_est_bissextile (annee: int) -> bool:

"""

Evalue si l'année est bissextile

Préconditions : annee $\geq$ 1582.

Exemples

\$\$\$ annee_est_bissextile(2000)
True

\$\$\$ annee_est_bissextile(2024)
True

\$\$\$ annee_est_bissextile(2023)
False

\$\$\$ annee_est_bissextile(2019)
False.

"""

return (annee % 4 == 0 and annee % 100 != 0) OR annee % 400 == 0.