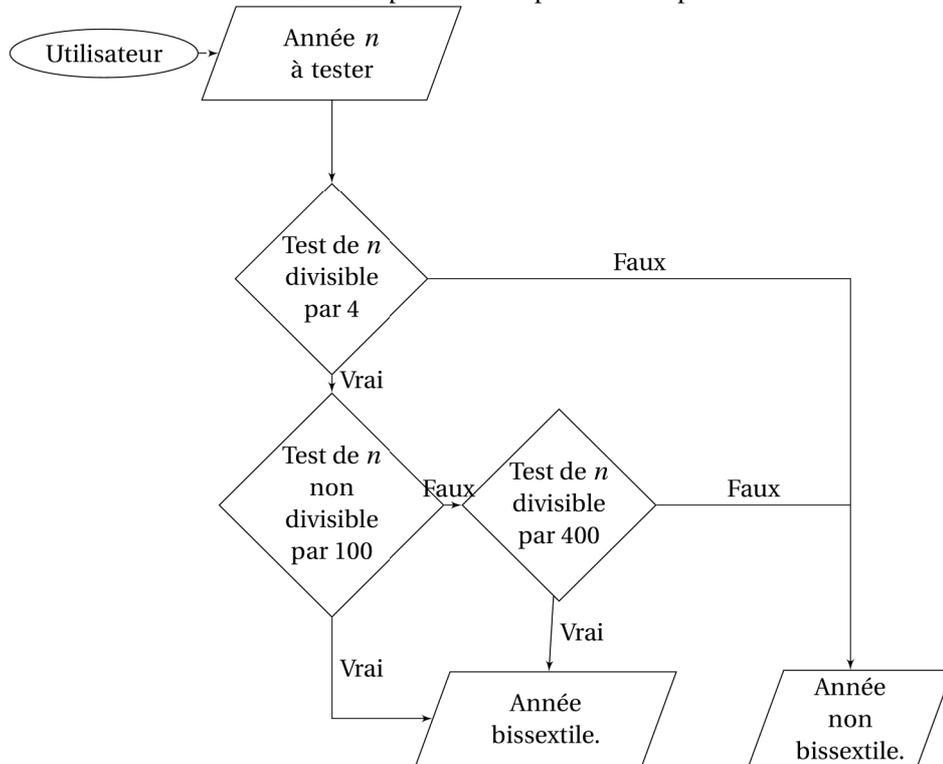


. Correction du TP02

1. $4|1900$ et $100|1900$, donc 1900 n'est pas bissextile.
 $4 \nmid 1983$ et $400 \nmid 1983$, donc 1983 n'est pas bissextile.
 $400|2000$, donc 2000 est bissextile.
 $4|2016$ et $100 \nmid 2016$, donc 2016 est bissextile.
 $4 \nmid 2018$ et $400 \nmid 2018$, donc 2018 n'est pas bissextile.
 $4|2020$ et $100 \nmid 2020$, donc 2020 est bissextile.
 $4|2100$ et $100|2100$, donc 2100 n'est pas bissextile.
2. (a) : N.B. : Un millésime non divisible par 4 ne sera pas divisible par $400 = 4 \times 100$.



(b) Algorithme en langage naturel :

```

1 Entrées : Demander l'année à tester à l'utilisateur
2 début
3   Stocker ce nombre dans la variable n, sous forme d'entier.
4   si n est divisible par 4 alors
5     si n n'est pas divisible par 100 alors
6       Afficher : "Année bissextile"
7     sinon
8       si n est divisible par 400 alors
9         Afficher : "Année bissextile"
10      sinon
11        Afficher : "Année non bissextile"
12    sinon
13      Afficher : "Année non bissextile"
  
```

Algorithme 3 : Algorithme en "langage naturel"

(c) :

Une première correction qui suit à la lettre notre diagramme :

Les commentaires (précédés du symbole#) ne font pas partie du programme.

```

n = int(input("Entrez l'année à tester"))
if n % 4 == 0:    #Test de n divisible par 4
    if n % 100 != 0:    #Test de n non divisible par 100
  
```

```

        print("Année bissextile")
    else:
        if n % 400 == 0:    #Test de n divisible par 400
            print("Année bissextile")
        else:
            print("Année non bissextile")
else:
    print("Année non bissextile")

```

Une deuxième correction avec "elif" (else if / sinon si), et l'utilisation d'un booléen :

```

n = int(input("Entrez l'année à tester"))
bissextile = False
if n % 4 != 0:
    bissextile = False
elif n % 100 == 0:
    if n % 400 == 0:
        bissextile = True
    else:
        bissextile = False    #Ce bloc "elif" est équivalent à : bissextile = n %
400 == 0
else:
    bissextile = True
if bissextile == True:
    print("Cette année est bissextile")
else:
    print("Cette année n'est pas bissextile")

```

3. (a) Exemples d'expressions booléennes :

```

3*x+1 == 2
7*x != 3
2*a < 4
x % 5 == 0 (Cette dernière expression signifie "x est divisible par 5")

```

(b) Trois autres exemples :

```

x == 1 or x == 8
x != 2 and x == 4
(x*x == 16 or x*x == 25) and (x != -4 and x != -5)

```

(c) $n \% 400 == 0$

4. (a) $(\text{annee} \% 4 == 0 \text{ and } \text{annee} \% 100 != 0) \text{ or } (\text{annee} \% 400 == 0)$

(b) Nouvelle version du programme :

```

n = int(input("Entrez l'année à tester"))
bissextile = (n % 4 == 0 and n % 100 != 0) or (n % 400 == 0)
if bissextile == True:
    print("Cette année est bissextile")
else:
    print("Cette année n'est pas bissextile")

```

Correction de la question supplémentaire :

```

def bissext(annee):
    return (annee % 4 == 0 and annee % 100 != 0) or (annee % 400 == 0)

def nb_bissext(debut,n):
    compt = 0
    for k in range(n):
        annee = debut + k
        if bissext(annee):
            compt = compt + 1
    return compt

debut = int(input("A partir de quelle année voulez-vous compter les années bissextiles ?"))
n = int(input("Sur combien d'années voulez-vous compter ?"))

resultat = nb_bissext(debut,n)

print("Il y a {} année(s) bissextiles dans les {} années qui suivent {}".format(resultat,n,debut))

```