

Correction TD-TP 1

Algorithmique du traitement des données

Raphaël Fournier-S'niehotta, fournier@cnam.fr

Octobre 2016

1 Algorithmique

1.1 Méthodologie de conception

Faire les étapes 1, 2 et 3 et 4 pour chacun des problèmes suivants :

1.2 Note d'UE

Calculer la note finale d'une unité d'enseignement étant données la note du partiel et celle de l'examen et sachant que :

- une note inférieure à 7 à l'examen est éliminatoire
- si la note d'examen est supérieure à 7, la note finale est la moyenne des deux notes si elle avantage l'étudiant. Sinon, la note finale est la note d'examen.

1. **entrée** : note d'examen (ne), note de partiel (np)

2. **sortie** : note finale (nf)

3. **tests** :

ne	np	nf	explication
6	qcq	6	ne < 7
8	6	8	moyenne de ne et np < ne
8	12	10	moyenne de ne et np > ne
14	16	15	moyenne de ne et np > ne

4. **calcul des sorties à partir des entrées** :

```
si ne < 7 alors nf = ne
sinon faire la moyenne de ne et np (moy)
    si moy > ne alors nf = moy
    sinon nf = ne
```

1.3 Entier dans un intervalle

Tester si un entier appartient à l'intervalle donné par deux nombres entiers.

1. **entrée** : 1 entier a, 1 intervalle de 2 entiers (c et d)

2. **sortie** : un booléen (dans)

	a	[c,d]	dans	explication
3. tests :	12	[10,24]	True	$10 < 12 < 24$
	6	[10,24]	False	$6 < 10$
	30	[10,24]	False	$30 > 24$

4. calcul des sorties à partir des entrées :

si $a < c$ ou $a > d$ alors `dans` = False
sinon `dans` = True

1.4 Plus grand de 3 entiers

Calculer puis afficher le plus grand parmi trois nombres entiers.

1. **entrée** : 3 entiers, a b et c

2. **sortie** : un entier, max

	a	b	c	max
3. tests :	6	10	45	45
	6	7	4	7
	6	3	2	6

4. calcul des sorties à partir des entrées :

si $a > b$ alors `max` = a sinon `max` = b
si $c > \text{max}$ alors `max` = c

1.5 Tri de 3 entiers

Tester si trois nombres entiers sont triés (dans l'ordre croissant ou décroissant).

1. **entrée** : 3 entiers a b c

2. **sortie** : booleen ord

	a	b	c	ord
3. tests :	6	10	45	True
	6	7	4	False
	6	4	8	False
	6	3	2	True

4. calcul des sorties à partir des entrées :

si $((a > b \text{ et } b > c) \text{ ou } (a < b \text{ et } b < c))$
alors `ord` = True
sinon `ord` = False

1.6 Année bissextile

Tester si une année est bissextile. On sait qu'une année divisible par 4 est bissextile sauf si elle est divisible par 100, cependant les années divisibles par 400 sont également bissextiles.

1. **entrée** : un entier (annee)

2. **sortie** : un booleen bissextile

annee	bissextile	explication
2000	True	divisible par 4, 100 et 400
1900	False	divisible par 4 et 100 mais pas 400
2016	True	divisible par 4 mais pas par 100
2017	False	non divisible par 4

3. tests :

4. calcul des sorties à partir des entrées :

```

si a divisible par 4
alors si a divisible par 100
    alors si a divisible par 400
        alors bissextile = True
        sinon bissextile = False
    sinon bissextile = True
sinon bissextile = False

```

2 Terminologie

Soit le morceau de code suivant : $(x * 3) + 1$

- quel nom donne-t-on à ce morceau de code tout entier ? **une expression**
- quels sont les opérateurs ? **+ et ***
- quels sont les opérands de * ? **x et 3**
- quels sont les opérands de + ? **(x*3) et 1**
- quel genre de chose est x dans ce morceau de code ? **une variable**
- quel(s) type(s) a ou peut éventuellement avoir x ? **x doit pouvoir se multiplier par 3, donc un type numérique, entier ou flottant**
- quel est le type de 3 et 1 ? **entier**
- quel est le type de $(x * 3) + 1$? **le même que celui de x**
- quelle est la valeur de $(x * 3) + 1$? **sans connaître la valeur de x, on ne sait pas**

3 Traces d'exécution

3.1 Somme 1

```

1 x = 3
2 y = 4
3 z = x+y
4 print("la somme de"+ x + "et de"+ y + "est "+z)

```

	x	y	z	clavier	ecran
1	3	X	X		
2	3	4	X		
3	3	4	7		
4	3	4	7		la somme de 3 et 4 est 7

3.2 Somme 2

```
1 x = 3
2 y = 4
3 x = x+y
4 print("la somme de"+ x + "et de"+ y + "est "+x)
```

	x	y	clavier	ecran
1	3	X		
2	3	4		
3	7	4		
4	3	4		la somme de 7 et 4 est 7

3.3 Somme 3

```
1 x = 3
2 y = 4
3 print("la somme de"+ x + "et de")
4 x = x+y
5 print(y + "est "+x)
```

	x	y	clavier	ecran
1	3	X		
2	3	4		
3	3	4		la somme de 3 et de
4	7	4		
5	7	4		4 est 7

4 Opérateurs booléens

```
print(True and False)
a = True
b = True
c = True
print(b and c)
print(not a)
print((not a) or (b and c))
```

affichage :

```
False
True
False
True
```

```
a = "x"
b = 2
print("Valeur de 5 > 3: ")
print(5>3)
print("Valeur de a == 'b': ")
print(a == 'b')
print("Valeur de (b>=0) && (b<=100): ")
print((b>=0) && (b<=100))
```

affichage :

Valeur de `5 > 3`:

True

Valeur de `a == b`:

False

Valeur de `(n>=0) && (n<=100)`:

True
