

Initiation à la programmation

Première partie du cours

Exercices

Exercice 1 : Les systèmes de numération

Quelques puissances de 2 :

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}
1	2	4	8	16	32	64	128	256	512	1024	2048

Quelques puissances de 8 et de 16 :

8^0	8^1	8^2	8^3	8^4	16^0	16^1	1		
1	8	64	512	4096	1	16	256	4096	

Chaque colonne du tableau suivant représente un système de numération (celui repris en tête de colonne). Remplissez les cases laissées vides dans ce tableau ? N'oubliez pas de donner la base de chaque système !

systèmes	binaire	octal	décimal	hexadécimal
bases?				
	1101010			
			14,5	

Exercice 2 : Théorie

a) Imaginons un nombre entier non signé codé sur 4 bits.

Combien y a-t-il de possibilités ? ...

Quel est le minimum ? ...

Quel est le maximum ? ...

Imaginons un nombre entier signé (complément-à-2) codé sur 4 bits.

Combien y a-t-il de possibilités ? ...

Quel est le minimum ? ...

Quel est le maximum ? ...

b) Quels types « Python » utiliseriez-vous pour représenter :

- a. un entier? ...
- b. un réel? ...
- c. une chaîne de caractères? ...
- d. un booléen ? ...

c) Quel est l'équivalent du nombre ci-dessous en système décimal, sachant qu'il s'agit d'un nombre binaire **signé** en complément-à-2 codé sur 4 bits.

$$(1010)_{2 \text{ c-à-2}} \rightarrow (\quad)_{10}$$

d) Pour chaque ligne en Python ci-dessous, indiquez le résultat dans le cadre.

```
01 : print 10 / 4
02 : print 10.0 / 4
03 : print 10.0 // 4
04 : print 10.0 / "4"
```

01 :

02 :

03 :

04 :

Exercice 3 : Logique combinatoire

Soit F une fonction incomplètement définie :

$$F(d,c,b,a) = \sum [0,2,4,6,7,8,10,12,13,14,15] + \prod [1,3,9]$$

Complétez la table de vérité ci-dessous. Ensuite, représentez un diagramme de Karnaugh correspondant et déterminez l'expression la plus simple possible pour F.

n	d	c	b	a	F
00	0	0	0	0	.
01	0	0	0	1	.
02	0	0	1	0	.
03	0	0	1	1	.
04	0	1	0	0	.
05	0	1	0	1	.
06	0	1	1	0	.
07	0	1	1	1	.
08	1	0	0	0	.
09	1	0	0	1	.
10	1	0	1	0	.
11	1	0	1	1	.
12	1	1	0	0	.
13	1	1	0	1	.
14	1	1	1	0	.
15	1	1	1	1	.

Exercice 4 : Codages et addition binaire

A. Représentez dans le tableau les nombres X et Y sur 8 bits (complément-à-2).

Faites ensuite l'addition de ces nombres et indiquez le résultat dans la dernière ligne du tableau.

X : (45) ₁₀							
Y : (-114) ₁₀							
Résultat							

Exercice 5 : Questions à choix multiples

REPORTEZ VOS REPONSES SUR LA PAGE DE GARDE !!!

Choisissez la bonne réponse pour chacune des questions qui suit. Seules les réponses reportées sur la page de garde seront prises en compte.

- Bonne réponse → +1 [pts]
- Absence de réponse → +0,25 [pts]
- Mauvaise réponse → 0 [pts]

A. Quel est le résultat du fragment de code suivant :

```
01: a,b= "2","3"
02: c,d = 4,5
03: e = a+b
04: f = c+d
05: print str(e) + str(f)
```

- a. 2345
 - b. 239
 - c. 59
 - d. 14

B. Quel est le résultat du fragment de code suivant :

```
01: if not(False):
02:     if 1 != 1:
03:         print "1"
04:     elif 5/2 == 2:
05:         print "2"
06:     elif 5//2 == 2:
07:         print "3"
08: else:
09:     print "4"
```

- a. 1
 - b. 2
 - c. 3
 - d. 4

C. Quel est le résultat du fragment de code suivant :

```
01: fruit,legume = 'melon','topinambour'
02: a,b,c = 1,2,3
03: if a % 2 == b:
04:     print "%i %s !" % (a,fruit)
05: elif c % 2 == a:
06:     if b % 2 == a:
07:         print "%i %s !" % (b,fruit)
08:     print "%i %s !" % (a,legume)
09: else:
10:     print "%i %s !" % (c,legume)
```

- a. 3 topinambour !
- b. %i %s ! 2 melon
%i %s ! 1 topinambour
- c. 2 melon !
1 topinambour !
- d. 1 topinambour !

D. Que produira le morceau de code suivant :

```
01: a = 0
02: # a = a+1
03: ""
04: a = a+2
05: a = a+3
06: ""
07: a = a+4 # a = a+5
08: print a
```

- a. 0
- b. 4
- c. 5
- d. 15

Exercice 6 : Correction de code

Le programme python ci-dessous qui répond aux spécifications données comporte environ 12 erreurs. Corrigez-le pour que ce programme soit syntaxiquement correct pour un interpréteur python 2.7 et qu'il produise les résultats attendus par les spécifications. Des « corrections » de votre part qui entraînent de nouvelles erreurs seront pénalisées.

Un programme compare 3 nombres, soit les nombres 5, 7 et un troisième nombre reçu au clavier. Le programme affiche un message rapportant quel est le maximum de ces nombres. S'il y a des nombres identiques parmi ces 3 nombres, le programme affiche également combien il y a de nombres identiques.

CORRIGEZ DIRECTEMENT SUR LE CODE DANS UNE AUTRE COULEUR QUE LE ROUGE.

```

1 messageDemande = 'Entrez l'entier numéro %i : '
2 messageReponse = "Le maximum est : "
3 messageEgalite = "Il y a %i nombres égaux."
4 maximum = None
5 nbEgalite = 0
6
7 a = 5, b = 7
8 c = raw_input(messageDemande % (3))
9
10 if (a >= b) and (a >= c)
11     maximum = a
12 elif (b >= a) and (b >= c):
13     maximum = "b"
14 else (c >= b) and (c >= a):
15     maximum == c
16
17 if (A == B == C):
18     nbEgalite = 3
19 elif (a=b or a==c or b==c):
20     nbEgalite = 2
21
22 print messageReponse + maximum
23 if nbEgalite:
24 print messageEgalite % (nbEgalite)

```

Exercice 7 : Algorithme et écriture de code

Donner l'**algorithme** et le **code python** pour le programme décrit ci-dessous.

- Le programme reçoit 2 cotes /20 au clavier. On considère ici que les cotes introduites par l'utilisateur sont correctes.
- Il affiche le nombre de cotes inférieures à 10.
- Il calcule la moyenne, puis affiche cette moyenne et affiche si l'étudiant a raté ou réussi. Un étudiant a réussi s'il a au moins 10 de moyenne et aucune note inférieure à 10.

Exemple de résultat attendu avec 10 et 10 comme cotes introduites :

Il y a 0 cote(s) en-dessous de 10.

moyenne : 10.00 ; réussi.

Exemple de résultat attendu avec 4.5 et 5.5 comme cotes introduites :

Il y a 2 cote(s) en-dessous de 10.

moyenne : 5.00 ; raté.