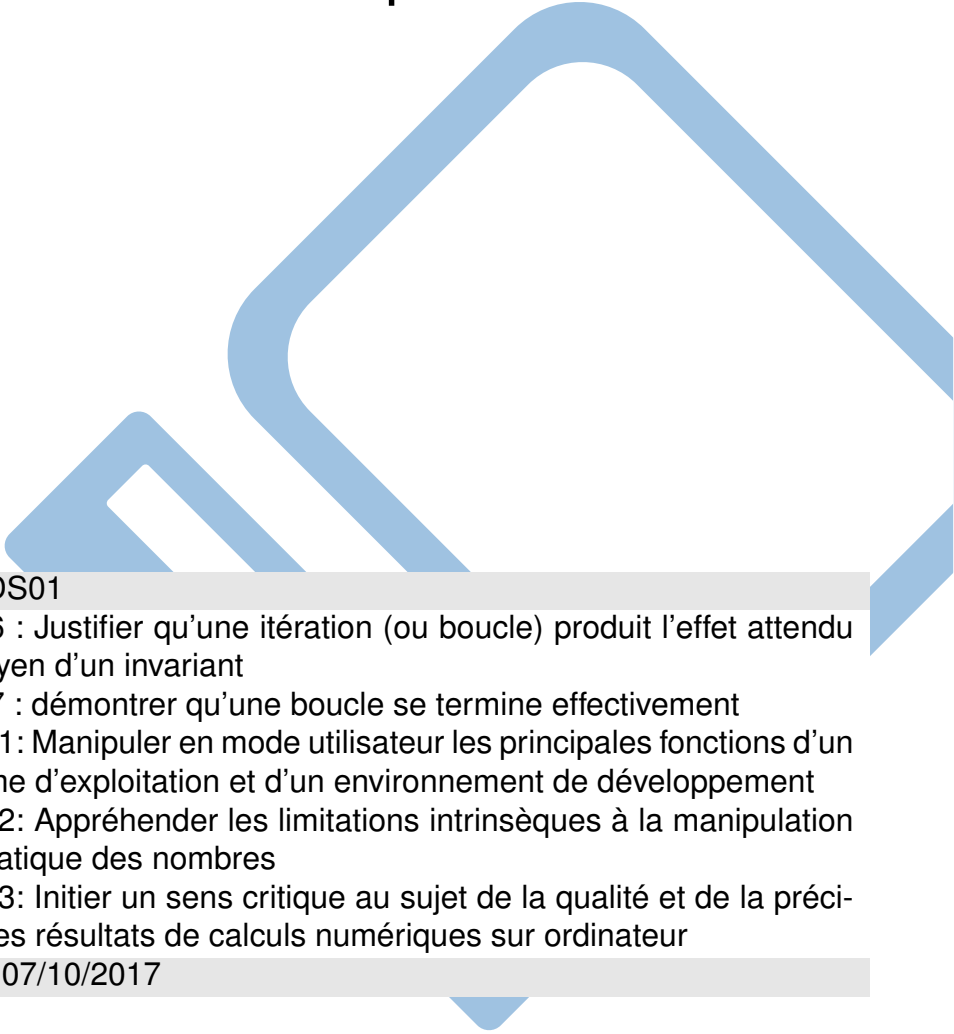


Avec Correction

DS01 Informatique



Référence	S01- DS01
Compétences	Alg-C6 : Justifier qu'une itération (ou boucle) produit l'effet attendu au moyen d'un invariant Alg-C7 : démontrer qu'une boucle se termine effectivement Déc-C1: Manipuler en mode utilisateur les principales fonctions d'un système d'exploitation et d'un environnement de développement Déc-C2: Appréhender les limitations intrinsèques à la manipulation informatique des nombres Déc-C3: Initier un sens critique au sujet de la qualité et de la précision des résultats de calculs numériques sur ordinateur
Description	Fait le 07/10/2017

Les 3 parties sont à rendre sur des copies séparées.

1 Introduction

Question 1 : Écrire sur le diagramme de Contexte donné en document réponse le nom des composants de l'unité centrale.

Question 2 : Écrire sous la forme d'un mot de 32 bits respectant la norme IEEE 754 (signe, exposant, mantisse), le float $\frac{14}{9} = 1,5555\dots$

Question 3 : Traduire en float le mot de 32 bits suivant.

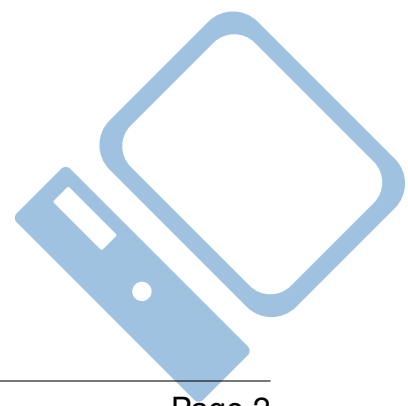
[illegible]

2 Emploi du temps

Une classe est organisée comme suit :

- un groupe de TD a cours une semaine sur 2 :
 - Semaine N : groupe 'G1',
 - Semaine N+1 : groupe 'G2',
 - Semaine N+2 : groupe 'G1',...
- un groupe de TP a cours une semaine sur 3 :
 - Semaine N : groupe 'GA',
 - Semaine N+1 : groupe 'GB',
 - Semaine N+2 : groupe 'GC',
 - Semaine N+3 : groupe 'GA',...

Le programme python permet d'afficher un tableau pour lequel chaque ligne correspond à une semaine, la première colonne le numéro de la semaine, la seconde le groupe de TD et la troisième le groupe de TP.



```
1  semaine=1
2  while semaine < 45: #Tant que semaine<45, exécuter la suite
3      #Groupe TD
4      if semaine%2==0:
5          groupe_td='G1'
6      else:
7          groupe_td='G2'
8
9      #Groupe TP
10     if semaine%3==2:
11         groupe_tp='GA'
12     elif semaine%3==1:
13         groupe_tp='GB'
14     else:
15         groupe_tp='GC'
16     print semaine,groupe_td,groupe_tp
17     semaine=semaine+1
```

Question 4 : D'après le code présenté quelle sera la première ligne affichée ? Quelle sera la dernière ?

Un problème d'organisation fait que lorsque le groupe de TD 'G1' et le groupe de TP 'GC' ont cours la même semaine, il faut que les TP aient lieu le lundi.

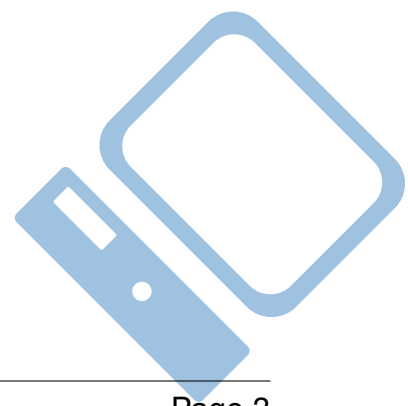
Question 5 : Proposer une modification pour que programme affiche au bout de la ligne correspondante le texte 'TP le lundi'. L'affichage devra correspondre à l'exemple suivant :

```
35 G2 GA
36 G1 GC TP le lundi
```

Question 6 : On souhaite ajouter un quatrième groupe de TP, proposer une modification du code pour que cela soit possible. La seule contrainte concernant l'alternance des groupes est que la combinaison qui impose les TP le lundi n'intervienne jamais.

3 Calcul de 2^n

L'objectif du code suivant est de calculer 2^n .



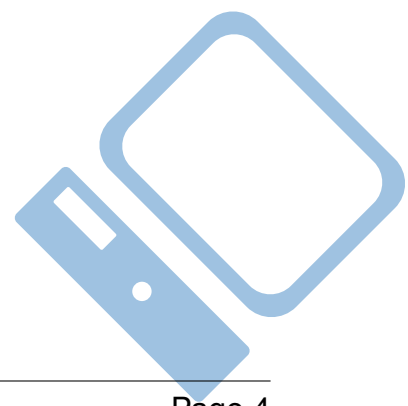
```
1 def puissance_deux(n):  
2     y, n_loc = 1,n  
3     while n_loc > 0:  
4         y = y*2  
5         n_loc = n_loc-1  
6     return y
```

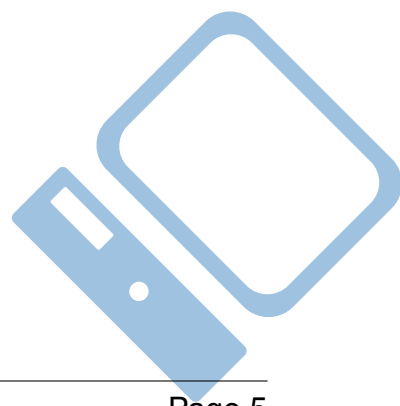
Question 7 : Écrire pour chaque itération de la boucle les valeurs de y et de n_loc pour le calcul de `puissance_deux(4)`.

- $y_0=...$ et $n_loc_0=...$,
- $y_1=...$ et $n_loc_1=...$,
- $y_2=...$ et $n_loc_2=...$,
- ...,

Question 8 : Que se passe-t-il si n est négatif ?

Question 9 : Proposer une modification pour calculer 2^n avec ($n < 0$) ?

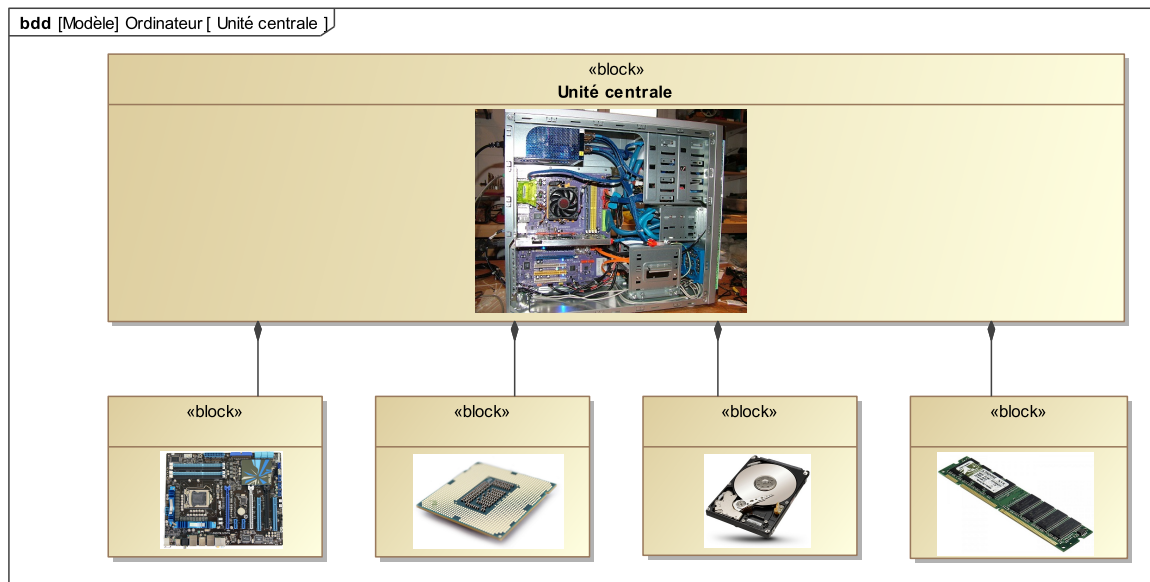




4 Document réponse

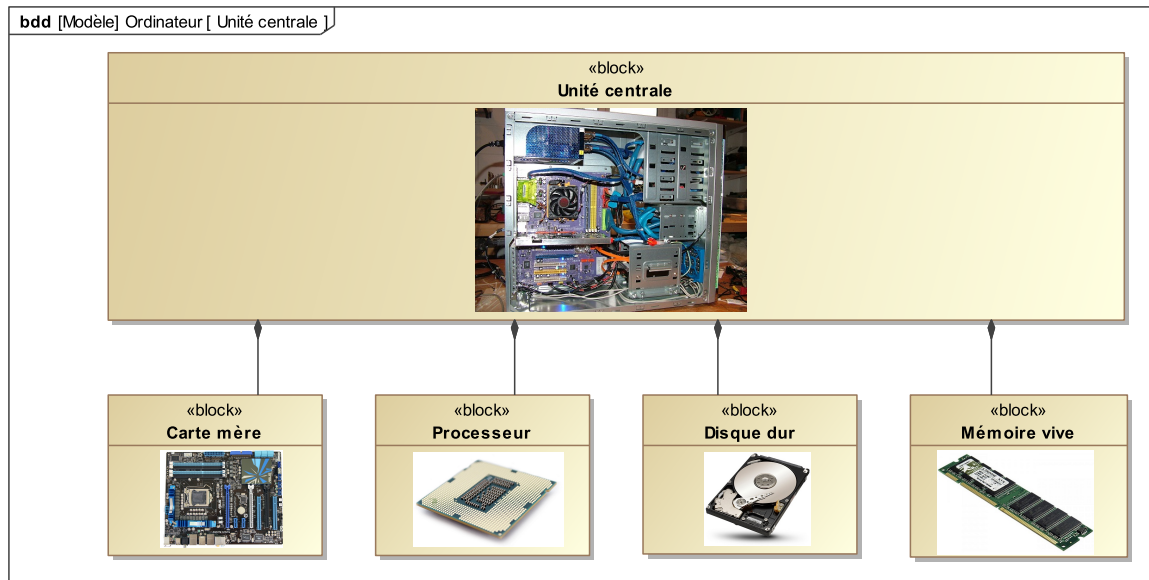
Nom :

Prénom :



1 Correction

Question 1 :



Question 2 : Le nombre à traduire est $\frac{14}{9} = 1,5555$.

$$0,555.. \times 2 = 1,11.. = 1 + 0,11..$$

$$0,111.. \times 2 = 0,22.. = 0 + 0,22..$$

$$0,222.. \times 2 = 0,44.. = 0 + 0,44..$$

$$0,444.. \times 2 = 0,88.. = 0 + 0,88..$$

$$0,888.. \times 2 = 1,77.. = 1 + 0,77..$$

$$0,777.. \times 2 = 1,55.. = 1 + 0,55..$$

$$0,555.. \times 2 = 1,11.. = 1 + 0,11..$$

$$0,111.. \times 2 = 0,22.. = 0 + 0,22..$$

...

On remarque une récurrence dans l'écriture du $1,555_{10}$ en binaire : $1,555_{10} = 1,10001110_{10}$

$$1, \underbrace{10001...01}_{23\text{bits}} \times 2^0$$

— Signe = 0,

— Mantisse : $\underbrace{10001...01}_{23\text{bits}}$,

— Exposant : $127 + 0 = 127_{10} = 00111111_2$

S	Exposant								Mantisse																							
0	0	0	1	1	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1

Question 3 : Exposant : $10001110 = 2 + 4 + 8 + 128 = 142$

Exposant simple : $142 - 127 = 15$

Mantisse : $1,0001010001100000000000_2 \times 2^{15}$

Nombre $100010100011_2 \times 2^4 = (2048_{10} + 128_{10} + 32_{10} + 2_{10} + 1_{10}) \times 16 = 35376_{10}$.

Question 4 : L'affichage de la première ligne sera 1 G2 GB, la dernière 44 G1 GA.

Question 5 :

```
semaine=1

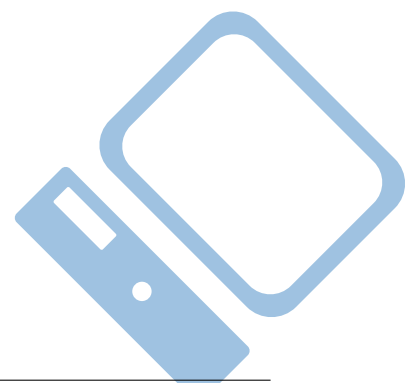
while semaine < 45:
    #Groupe TD
    if semaine%2==0:
        groupe_td='G1'
    else:
        groupe_td='G2'

    #Groupe TP
    if semaine%3==2:
        groupe_tp='GA'
    elif semaine%3==1:
        groupe_tp='GB'
    else:
        groupe_tp='GC'

    if groupe_td=='G1' and groupe_tp=='GC':
        print semaine,groupe_td,groupe_tp,'TP le lundi'
    else:
        print semaine,groupe_td,groupe_tp

    semaine=semaine+1
```

Question 6 : Pour que la combinaison n'intervienne pas, une possibilité est de contraindre le groupe 'G1' sur des semaines paires et le groupe 'GC' sur des semaines impaires.




```
semaine=1

while semaine < 45:
    #Groupe TD
    if semaine%2==0:
        groupe_td='G1'
    else:
        groupe_td='G2'

    #Groupe TP
    if semaine%4==0:
        groupe_tp='GA'
    elif semaine%4==1:
        groupe_tp='GB'
    elif semaine%4==3:
        groupe_tp='GC'
    else:
        groupe_tp='GD'

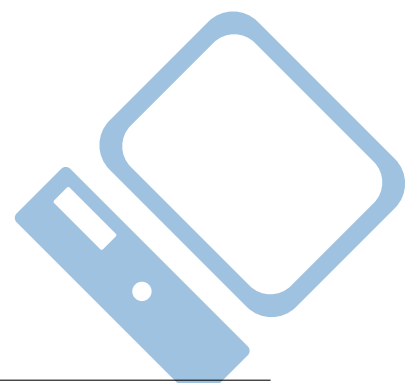
    semaine=semaine+1
```

Question 7 : Écrire pour chaque itération de la boucle les valeurs de y et de n_loc , pour le calcul de `puissance_deux(4)`.

- $y_0=1$ et $n_loc_0=4$,
- $y_1=2$ et $n_loc_1=3$,
- $y_2=4$ et $n_loc_2=2$,
- $y_3=8$ et $n_loc_3=1$,
- $y_4=16$ et $n_loc_4=0$.

Question 8 : Si n est négatif, n_loc est négatif et $y=1$. L'étape `while n_loc > 0` rejette ce cas de figure : on n'entre pas dans la boucle `while`. Donc y n'est pas modifié. Le programme retourne 1.

Question 9 :



Correction

```
def puissance_deux(n):  
    y, n_loc = 1, n  
    if n>0:  
        while n_loc > 0:  
            y=y*2  
            n_loc = n_loc-1  
    else:  
        while n_loc < 0:  
            y=y/2.  
            n_loc = n_loc+1  
    return y
```

