I11: Contrôle terminal - session 1
Licence 1 MATHS, PC, SI

Janvier 2018 (semestre 1) - Durée : 2h00

- Tous les documents, calculatrices et appareils de communication sont interdits.
- Le barême est donné à titre indicatif
- Tous les scripts devront être clairement indentés et les noms de variables choisis de façon appropriée.
- Seules les instructions et fonctions internes à Python **vues en cours** sont autorisées.

EXERCICE 1. (2 points)
On considère les déclarations de variables suivantes:

```
nb = "12345"
L=[nb, 3.14, ["je","tu","il","nous","vous","ils"], (-1,2)]
```

Donner le **type** et la **valeur** des expressions suivantes:

```
L[0],       L[1]+str(L[0][1]),    (2,-1)+L[3],
L[2][1:4],  L[-2][1::2],          L[0][:2]*L[-1][-1],
```

EXERCICE 2. (1 points) Écrire un script qui demande deux nombres entiers à l'utilisateur et affiche toutes les valeurs de la suite $U_0 = 1, U_{n+1} = \frac{U_n^2}{2} - 2U_n + 1$ pour n compris entre ces deux valeurs (incluses).

Exemple:

```
>>>
 Saisir un entier: 1
 Saisir un entier: 3
-0.5
2.125
-0.9921875
```

Exercice 3. (2 points)

On considère le script suivant :

```
n = int(input())
crible = [1]*n
i=2
while i<= n**0.5:
  if crible[i]==1:
    j=i*i
    while j<n:
      crible[j]=0
      j=j+i
  i=i+1
for i in range(2,len(crible)):
  if crible[i]==1:
    print(i)
```

Faire une table des valeurs de ce script pour l'entrée `n=16` sur le modèle suivant:

i	j	crible[i]==1	j<=n	ECRAN

Exercice 4. (4 points)

1. Écrire un script qui calcule la moyenne des nombres strictement positifs d'une part et la moyenne des nombres strictement négatifs d'autre part d'une série de nombres saisie par l'utilisateur; la saisie s'arrête quand le nombre 0 est rentré.

   Exemple:

```
>>>
 Saisir un nombre: 2
 Saisir un nombre: 10
 Saisir un nombre: -3
 Saisir un nombre: 9
 Saisir un nombre: -6
 Saisir un nombre: 0
 Moyenne positifs: 7.0
 Moyenne negatifs: -4.5
```

2. Écrire un script qui demande 2 chaînes de caractères `ch1` et `ch2` à l'utilisateur, la première de longueur quelconque et la deuxième de longueur 2 (inutile de faire la vérification dans le script), et affiche le nombre d'occurences de `ch2` dans `ch1`. Par exemple

```
>>>
  ch1 = ceci est certainement un exemple recent
  ch2 = ce
  nombres d'occurences: 3

>>>
  ch1 = aaa
  ch2 = aa
  nombres d'occurences: 2
```


2

3. On considère la liste prédéfinie `L=[1,2,3,4,5,6,7,8,9,10]`. Écrire un script qui demande un nombre n à l'utilisateur et effectue une rotation de n cases sur la droite des éléments de la liste. Par exemple, pour $n = 2$ la liste deviendra `[9,10,1,2,3,4,5,6,7,8]`.

4. On définit une matrice comme étant une liste de listes de nombre. Par exemple:

$$M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = \texttt{[[1,2,3],[4,5,6],[7,8,9]]}.$$

On dit qu'une matrice est creuse si strictement plus de la moitié de ses coefficients sont nuls. Écrire une fonction `est_creuse(A)` qui retourne `True` si la matrice `A` est creuse et `False` sinon. Par exemple:

```
>>> est_creuse([0,0,1],[1,0,0],[1,1,0])
 True
>>> est_creuse([0,1],[1,0])
 False
```

Exercice 5. (5 points)
On considère que le script et les fonctions suivants sont écrits dans le même fichier.

1. La distance entre deux mots est le nombre de lettres en lesquelles ils diffèrent. Par exemple la distance entre `caste` et `vaste` vaut 1, celle entre `part` et `partir` vaut 2 et celle entre `crypte` et `egyptien` vaut 5. Écrire un script qui retourne la distance entre deux mots saisis par l'utilisateur. Écrire une fonction `distance(ch1, ch2)` qui retourne la distance entre les deux chaînes de caractères `ch1` et `ch2`.

2. Écrire une fonction `decoupe(ch)` retourne une liste contenant tous les mots de la chaine `ch`. On suppose que `ch` est une suite de mots séparés uniquement par un ou plusieurs espaces.

```
>>> decoupe("ceci est un exemple     avec des espaces")
 ["ceci","est","est","un","exemple","avec","des","espaces"]
```

3. On considère une chaîne de caractères prédéfinie `vocabulaire` ayant le même format que la chaîne `ch` précédente. Écrire un script qui affiche les deux mots les plus proches de cette chaîne au sens de la distance de la question 1. Par exemple pour la chaîne `vocabulaire = "mais ou et donc or ni car"` le script devra afficher ou et `or`.

EXERCICE 6. (5 points)

On considère que les fonctions suivantes sont écrites dans le même fichier. Une équipe de football sera représentée en Python par un tuple contenant son nom, une liste de caractères représentant ses résultats ("D" pour défaite, "N" pour nul et "V" pour victoire) et son nombre de buts marqués. Par exemple (après 5 matchs) `("Toulon", ["V","D","N","N","V"], 12)`. Un championnat sera représenté par une liste d'équipes.

1. Écrire une fonction `nb_points(e)` qui retourne le nombre de points correspondant au résultat de l'équipe `e`. On considèrera qu'une victoire rapporte 3 points, un nul 1 point et une défaite 0 point.

2. Écrire une fonction `compare(e1,e2)` qui retourne 1 si l'équipe `e1` est devant `e2` au classement, 0 si elles sont à égalité et -1 sinon. Pour comparer deux équipes on commencera par regarder laquelle possède le plus de point puis, en cas d'égalité celle, qui a marquée le plus de buts.

3. Écrire une fonction `champion(c)` qui retourne le nom de la ou des équipes du championnat `c` première(s) au classement.

I11: Contrôle terminal
session 2
Licence 1 MATHS, PC, SI

Juin 2018 (session 2) - Durée : 2h00

- Tous les documents, calculatrices et appareils de communication sont interdits.
- Le barême est donné à titre indicatif
- Tous les scripts devront être clairement indentés et les noms de variables choisis de façon appropriée.
- Seules les instructions et fonctions internes à Python **vues en cours** sont autorisées.

Exercice 1. (2 points)
On considère les déclarations de variables suivantes:

```
pi=3.1415
L=["ex", pi, (1,2,3), ["4","5","6","7","8","9"]]
```

Donner le type et la valeur des expressions suivantes:

```
L[1],       L[0]+str(L[1]),     L[3][0]+L[3][-1][0],
L[2][:2],   L[3][1::2],         L[-1][0]*L[2][1],
```

Exercice 2. (2 points)
Écrire un script qui demande deux nombres entiers à l'utilisateur et affiche tous les nombres impairs entre ces deux nombres.

Exemple:

```
>>>
 Saisir un entier: 3
 Saisir un entier: 11
3
5
7
9
11
```


5

Exercice 3. (2 points)

On considère le script suivant

```
n=int(input("entrer un nombre ? "))
while (n>0):
  print(n%10)
  n = n//10
```

1. Faire une table des valeurs de ce script pour $n = 45021$

n	$n > 0$	Ecran

2. Écrire un script qui affiche la somme des chiffres décimaux d'un entier n lu en entrée.

Exercice 4. (4 points)

1. Écrire un script qui calcule et affiche la somme des nombres pairs et la somme des nombres impairs d'une liste d'entiers `list_nbr` prédéfinie. Par exemple, pour la liste `list_nbr=[1,2,4,9,2,7]` le script calculera `1+9+7=17` et `2+4+2=8`.
2. La distance entre deux mots est le nombre de lettres en lesquelles ils diffèrent. Par exemple la distance entre `caste` et `vaste` vaut 1, celle entre `part` et `partir` vaut 2 et celle entre `crypte` et `egyptien` vaut 5. Écrire un script qui retourne la distance entre deux mots saisis par l'utilisateur.

Exercice 5. (4 points)

On considère que le script et les fonctions suivantes sont écrits dans le même fichier.

1. Écrire une fonction `factorielle(n)` qui retourne $n! = 1 \times 2 \times 3 \times \cdots \times (n-1) \times n$.
2. Écrire une fonction `binomial(n,k)` qui retourne la valeur du coefficient binomial $\binom{n}{k} = \frac{n!}{k!(n-k)!}$.
3. Écrire un script qui demande un entier n à l'utilisateur et affiche le polynome $(X+1)^n = \binom{n}{0} + \binom{n}{1}X + \cdots + \binom{n}{n-1}X^{n-1} + \binom{n}{n}X^n$.

   Par exemple, pour $n = 4$ le script affichera:

```
>>>
4
(X+1)^4 = X^4 + 4X^3 + 6X^2 + 4X + 1
```


6

EXERCICE 6. (6 points)

On considère que le script et les fonctions suivantes sont écrits dans le même fichier. Un point du plan sera représenté par un tuple de deux flottants. Notons `P` un tel tuple, ses coordonnées x et y seront donc données respectivement par `P[0]` et `P[1]`. On définit la distance entre deux points $P_1 = (x_1, y_1)$ et $P_2 = (x_2, y_2)$ par

$$d(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.$$

1. Écrire l'instruction permettant d'importer la fonction `sqrt` du module `math`.

2. Écrire une fonction `Distance(P1,P2)` qui retourne la distance entre les points `P1` et `P2`.

3. On décide de représenter un cercle par une liste composé d'un point et d'un flottant représentant son rayon. Par exemple le cercle de centre $(1, 0)$ et de rayon 1.5 sera représenté par $C = [(1, 0), 1.5]$.

   Écrire une fonction `EstDansCercle(P,C)` qui retourne `True` si le point `P` est à l'intérieur du cercle `C` (c'est-à-dire qu'il est à distance du centre inférieure ou égale au rayon) et `False` sinon.

4. On considère une liste de cercle `list_cercles` et un point `P` prédéfinis. Écrire un script qui affiche le cercle de plus petit rayon de la liste contenant le point `P`.

7

L1 SI

I21: Algorithmique
Contrôle terminal - session 2

Aucun document ou appareil électronique n'est autorisé.

Exercice 1. (2 points)

1. Qu'est-ce qu'une instance d'un problème algorithmique?
2. Quelle est la complexité du tri par insertion?
3. Étant donné un tableau d'entiers non trié, indiquer quelle méthode est plus efficace pour rechercher un élément dans celui-ci:
   - utiliser l'algorithme de recherche séquentielle;
   - trier le tableau puis rechercher l'élément avec un algorithme de recherche dichotomique.

   Jusitifiez clairement votre réponse.

Exercice 2. (2 points)

Pour chaque paire de fonctions f et g, dire si $f = O(g)$, $g = O(f)$ et $f = \Theta(g)$ (donner seulement la réponse la plus précise).

1. $f(n) = n\log(n^3)$, $g(n) = \log(n)n^3$
2. $f(n) = (n+1)(n-1)(n+2)$, $g(n) = \frac{(n^2+1)(n^2-4)}{100}$
3. $f(n) = 2^n$, $g(n) = 2^{n-2}$
4. $f(n) = n + \sqrt{n}$, $g(n) = n + \log(n)$
5. $f(n) = n!$, $g(n) = n^2(n-2)!$
6. $f(n) = 1 + 2 + \dots \lfloor \frac{n}{2} \rfloor$, $g(n) = (\lfloor \frac{n}{2} \rfloor + 1) + (\lfloor \frac{n}{2} \rfloor + 2) + \dots + n$

Exercice 3. (3 points)

Faire une preuve d'arrêt et donner la complexité des deux boucles suivantes:

```
BOUCLE 1
  DEBUT
    i ← 1
    TQ i ≤ n FAIRE
      j ← 1
      TQ j*j ≤ n
        j ← j+1
      FTQ
      i ← i+1
    FTQ
  FIN
```

```
BOUCLE 2
  DEBUT
    i ← 1
    j ← n
    TQ i ≤ j FAIRE
      i ← i*2
      j ← ⌊j/2⌋
    FTQ
  FIN
```


8

EXERCICE 4. (3 points)

On dit qu'un tableau est trié en vague si pour tout élément du tableau (à l'exception du premier et du dernier) est soit plus grand soit plus petit que ses deux voisins. Par exemple les tableaux `[2,4,1,3,0]` et `[1,1,1,1]` sont triés en vague mais pas le tableau `[2,3,4,1]`.

1. Écrire un algorithme `EstTrie(T)` qui retourne `VRAI` si le tableau `T` (de taille n) est trié en vague et `FAUX` sinon.
2. Faire une analyse en meilleur et pire cas de la complexité de l'algorithme et donner sa complexité générale.

EXERCICE 5. (4 points)

On considère un tableau d'entiers `T` de taille `n` et un entier `x`. La distance d'un élément du tableau avec `x` est simplement la valeur absolue de la différence entre `x` et l'élément en question.

Exemple:

```
T = [3, 1, 5, 9, 4], x = 7
distance entre T[1] et x: |7 - 3| = 4
distance entre T[2] et x: |7 - 1| = 6
etc
```

1. Écrire un algorithme `MaxDistance(T,i,x)` qui retourne l'indice de l'élément de `T` qui maximise la distance avec `x` parmi les éléments compris entre l'indice i et l'indice n. Par exemple, avec les données précédentes `MaxDistance(T,1,x)` retournera 2 et `MaxDistance(T,3,x)` retournera 5.
2. Écrire un algorithme de tri `TriDistance(T,x)` qui tri les éléments du tableau en fonction de leur distance à l'entier `x`, du plus éloigné au plus proche.

EXERCICE 6. (4 points)

1. Écrire un algorithme de compléxité $\Theta(n)$ `NegatifPositif(T)` qui traite un tableau d'entiers de taille n et range tous les entiers négatifs ou nuls à gauche du tableau et tous les nombres strictement positifs à droite.
2. On considère un tableau rangé par l'algorithme précédent. Écrire un algorithme en $O(\log(n))$ qui retourne le nombre d'éléments positifs du tableau.

EXERCICE 7. (2 points)

On rappelle les procédures standards de manipulation des piles et des files:

- `Vide(P)`: retourne `VRAI` si la pile `P` est vide
- `Empiler(P,x)`: ajoute l'élément `x` au sommet de la pile `P`
- `Depiler(P)`: retourne l'élément au sommet de la pile et le supprime
- `Lire(P)`: retourne l'élément au sommet de la pile

1. Écrire un algorithme `DepilerMax(P)` qui permet de supprimer le plus grand entier de la pile `P` sans changer l'ordre des autres éléments. L'algorithme ne doit utiliser qu'une deuxième pile et un entier comme variables auxillaires.

Licence SI 1ère année - Examen MS21 - Durée : 2h00

Les calculatrices et téléphones portables sont interdits. Les documents sont interdits.

Question de cours. Soit f une application définie sur un intervalle ouvert $I \subset \mathbb{R}$, On suppose que f est de classe $\mathcal{C}^3$ sur I. Ecrire le développement limité à l'ordre 3 au point x_0 pour f.

Dans ce qui suit, on pourra utiliser sans les redémontrer, les développements limités (DL) en 0 suivants :

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + o(x^n)$$
$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots + (-1)^n \frac{x^{2n}}{(2n)!} + o(x^{2n+1})$$
$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + o(x^{2n+2})$$
$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} + \cdots + (-1)^{n+1} \frac{x^n}{n} + o(x^n)$$

Exercice 1.
(1) Calculer le DL à l'ordre 2 en 0 de $\cos(x^2)\,(2 + \sin(x))$.
(2) Calculer, en justifiant votre calcul, le DL à l'ordre 3 en 0 de $e^{\ln(1+x)}$ (on utilisera sans justifier les DL de $\ln(1+x)$ et de $e^{(x)}$ rappelés ci-dessus). En déduire $\lim_{x \to 0, x \neq 0} \frac{e^{\ln(1+x)} - 1 - x}{x^2}$. Quel est l'ordre minimal du DL de $e^{\ln(1+x)}$ que l'on pouvait calculer pour déterminer cette limite : ordre 0, ordre 1, ordre 2 ou ordre 3 ?
(3) Calculer le DL à l'ordre 3 en 0 de $\frac{\cos(x)}{e^x}$.

Exercice 2. A l'aide de la formule de Taylor, et en détaillant les calculs, calculer le développement limité à l'ordre 1 en 0 de $\sqrt{1+x}$. En déduire un développement asymptotique en $+\infty$ de

$$f(x) = \frac{2\sqrt{1+x^2}}{1 + x + \sqrt{1+x^2}}.$$

Exercice 3. Calculer les primitives suivantes en précisant leur domaine de définition.

$$(1) \int (3x+1)^2\,dx \qquad (2) \int \frac{\sin(x)}{3 - \cos(x)}\,dx$$
$$(3) \int x\cos(x^2)\,dx \qquad (4) \int x\ln(x)\,dx$$
$$(5) \int \frac{1}{1+x^2}\,dx \qquad (6) \int x\cos(x)\,dx$$

Exercice 4.
(1) Parmi les intégrales généralisées suivantes, dire, en justifiant, celles qui sont convergentes ou pas.

$$\int_1^{+\infty} \frac{\sin(2+x^4)}{x^2}\,dx, \quad \int_0^1 \frac{\sin(x^2)}{x^{5/2}}\,dx, \quad \int_{-\infty}^0 \frac{1-x}{\sqrt{4+x+x^4}}\,dx, \quad \int_0^{+\infty} \frac{|\ln(x)|^3 e^{-x}}{\sqrt{x}}\,dx,$$

(2) Soit $M > 0$ fixé. A l'aide d'une intégration par partie, calculer

$$\int_{-M}^0 (1-x)e^x dx.$$

En déduire la nature de l'intégrale généralisée $\int_{-\infty}^0 (1-x)e^x dx$ et la calculer.

Licence SI 1ère année - Examen MS21 -

Les calculatrices et téléphones portables sont interdits. Les documents sont interdits.

Question de cours. Soit f une application définie sur un intervalle ouvert $I \subset \mathbb{R}$. On suppose que f est de classe $\mathcal{C}^3$ sur I. Ecrire la formule de Taylor-Young à l'ordre 3 au point $x_0 \in I$ pour f.

On rappelle les développements limités (DL) en 0 suivants :

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + o(x^3)$$
$$\sin(x) = x - \frac{x^3}{3!} + o(x^3)$$
$$\ln(1+x) = x - \frac{x^2}{2} + o(x^2)$$

Exercice 1. *Dans cet exercice, on pourra utiliser sans justifier les DL rappelés ci-dessus.*
(1) A l'aide de la formule de Taylor, calculer, en détaillant les étapes, le DL en 0 à l'ordre 2 de $\cos(x)$.
En déduire le DL en 0 de $\cos(2x)$ à l'ordre 2, puis calculer $\lim_{x\to 0, x\neq 0} \frac{\cos(2x)-1}{x^2}$.
(2) Calculer le DL à l'ordre 2 en 0 de $\frac{\ln(1+x)}{e^x}$.
(3) Calculer, en justifiant votre calcul, le DL à l'ordre 3 en 0 de $e^{\sin(x)}$.

Exercice 2.
(1) Soit $T > 0$ fixé. A l'aide d'une intégration par partie, calculer

$$\int_0^T (1+x)e^{-x}dx.$$

En déduire la nature de l'intégrale généralisée $\displaystyle\int_0^{+\infty} (1+x)e^{-x}dx$ et la calculer.
(2) Donner les primitives de $\dfrac{1}{1+x^2}$. Calculer les primitives de $(2x-1)^4$
(3) Parmi les intégrales généralisées suivantes, dire, en justifiant, mais sans chercher à les calculer, celles qui sont convergentes ou pas,

$$\int_1^{+\infty} \frac{\cos(x^4)}{x^2}\,dx, \quad \int_{-\infty}^0 \frac{1-x}{\sqrt{4+x+x^4}}\,dx, \quad \int_0^1 \frac{\sin(x)}{x^{3/2}}\,dx$$

L1Si-

Examen de M22-Complement d'analyse- Mai 2018. Note sur 20.
N.B. : le cours est autorisé ; les livres, les ordinateurs et smartphones sont interdits

Exercice 1. Calculer les limites suivantes

–
$$\lim_{n\to\infty}\left(\frac{n+2}{n}\right)^{n^{-\frac{1}{2}}}$$

–
$$\lim_{n\to\infty}(n^{\sqrt{n}}-2^n)$$

(*Suggestion* : passez aux exponentiels)

–
$$\lim_{n\to\infty}\sqrt[n]{\frac{n}{n^2+1}}$$

–
$$\lim_{n\to\infty}\frac{n^5+\ln(\sqrt{n})+2^n}{4^n+n^4+\ln^5 n}$$

Exercice 2. Résoudre la suite récurrente suivante d'ordre 2 :

$$u_{n+2}-6u_{n+1}+9u_n=0;\ u_0=5;\ u_1=6.$$

Exercice 3. Etudier la convergence des séries suivantes :

–
$$\sum_{n=0}^{\infty}\frac{\sqrt{n}}{n+1}$$

–
$$\sum_{n=1}^{\infty}\frac{(n!)^3}{3^{n^3}}$$

–
$$\sum_{n=1}^{\infty}\frac{1}{n^{n^2}}$$

–
$$\sum_{n=1}^{\infty}\frac{n^3}{3^n}$$

Exercice 4. Etudier la courbe parametrée définie par

$$\gamma(t):\begin{cases} x(t)=\frac{1}{t}\\ y(t)=\frac{(t^2+1)}{t}\end{cases}$$

12

L1 SI

Examen de M22. Compléments de Math. Juin 2018. Note sur 20. Calculettes, ordinateurs et téléphones INTERDITS.

June 9, 2018

1. Calculez les limites suivantes:

   - $$\lim_{n\to\infty} \frac{\sin(n^2+1)}{n^2+1}.$$

   - $$\lim_{n\to\infty} \frac{\sqrt{n+2}-\sqrt{n+1}}{\sqrt{n+1}-\sqrt{n}}.$$

   - $$\lim_{n\to\infty} (\sqrt{n^4+n^2}-n^2).$$

2. Etudier la convergence des séries suivantes:

   - $$\sum_{n=1}^{\infty} \frac{4n^2-n+3}{n^3+2n}.$$

   - $$\sum_{n=1}^{\infty} \frac{n+\sqrt{n}}{2n^3-1}$$

   .

3. Pour quelles valeurs de x la série suivante converge?

   - $$\sum_{n=1}^{\infty} \frac{\sqrt{x^2+n}-|x|}{n^2}.$$

4. Calculez la suite récurrente:

$$u_{n+2}-\sqrt{2}\,u_{n+1}+u_n=0,\ \text{ avec } u_0=u_1=1.$$

5. Tracer la courbe paramétrée

$$x(t)=t^3-t$$

$$y(t)=t^4-5\,t^2+4$$

avec le tableau des variations et les justifications rélatives.

13

S21 - Architecture I

Examen de session 1 - Licence SI - Année 1

23 mai 2018

- Tous les documents, calculatrices et appareils de communication sont interdits -

Le barème est donné à titre indicatif. Durée : 2 h.

PARTIE 1

Cette partie est notée sur 10 points, temps indicatif : 60 mn.

EXERCICE 1. **Architecture** (2 pts)

Une image numérique est une matrice de points de couleurs (ou pixels). Soit une image *noire et blanche* ayant une définition de 1024×512 pixels.

(1) Combien faut-il de bits pour coder un pixel noir ou blanc ? En déduire le nombre d'octets nécessaire pour coder l'image.

(2) Combien de mot-mémoires occupe l'image en mémoire centrale dans le cas d'une architecture 32 bits ? Et dans le cas d'une architecture 64 bits ?

EXERCICE 2. **Ordinapoche** (5 pts)

On rappelle les instructions connues d'Ordinapoche :

Instruction	Signification
INP	lecture depuis le périphérique d'entrée
OUT	affichage sur le périphérique de sortie
CLA	mise à zéro d'ACC et addition
STO	stocke le contenu d'ACC à l'adresse fournie
ADD	addition
SUB	soustraction
SHT	décalage gauche puis droite de l'ACC
JMP	saut inconditionnel à l'adresse fournie
TAC	si ACC $\neq 0$ alors saut à l'adresse fournie
HRS	fin de programme

Écrire un programme Ordinapoche qui affiche le nombre d'occurence du chiffre 1 dans un nombre n présent à l'adresse 99.

Attention : l'Ordinapoche ne gérant que les entiers positifs, on doit être particulièrement vigilant lors de l'utilisation de l'opérateur de soustraction.

La présentation ci-dessous doit être respectée.

Adr.	Instruction	Commentaires
00	CLA n	initialisation de l'ACC à n

Exercice 3. Base de numération et codage (3 pts)

(1) Donner la représentation binaire (en base 2) de l'hexadécimal $N = (FAC)_h$. Donner la représentation octale (en base 8) de N. Quelle est la taille en base 8 de N ?

(2) Donner les représentations en base 2 de $2^p - 1$ et de 2^p.

(3) Donner le codage sur 6 bits de 17. Donner le codage sur 6 bits de +17. Donner le codage sur 6 bits de -17 en complément logique.

(4) Donner l'interprétation en décimal de la séquence binaire 101.

	non signé	signe + VA	compl. logique	compl. arithmétique
101				

L1 SI

Partie 2

Cette partie contient 2 exercices et est notée sur 10 points. Temps indicatif : 60 mn

Tous les documents, calculatrices et appareils de communication sont strictement interdits.

Exercice 1 **(4 points)**

On veut réaliser un circuit ayant comme entrée un nombre positif A codé en binaire naturel sur 3 bits (A_0 est le LSB, ou bit de poids faible, de A), et une sortie B qui donne l'opposé de A dans le codage arithmétique (ou complément à 2).

Par exemple, si $A = (5)_{10}$, alors $B = (-5)_{10}$.

Questions :

1. Quelles valeurs décimales peut prendre A ? Quel codage correspond à chacune de ces valeurs ? (faire une table).
2. Quelles valeurs décimales peut prendre B ? Combien de bits sont nécessaires pour coder B (B_0 est le LSB de B) ? Quel codage correspond à chacune des valeurs que peut prendre B ? (compléter la table précédente).
3. En déduire les équations logiques simplifiées donnant les B_i en fonction des A_k. Utiliser, dès que possible, l'opérateur OU-exclusif !
4. Faire le schéma logique correspondant (un schéma propre est exigé !).

Exercice 2 **(4 points)** : On veut réaliser un circuit dont les entrées sont A, B et C, la sortie un afficheur 7 segments, et fonctionnant sur le principe suivant :

- ✓ si $A = B = C = 0$: on affiche 0,
- ✓ si $A = 1$: on affiche 3, quelles que soient les valeurs de B et C,
- ✓ si $A = 0$ et $B = 1$: on affiche 2, quelle que soit la valeur de C,
- ✓ sinon on affiche 1.

Les segments de l'afficheur sont codifiés sur le schéma suivant :

Questions :

1. Faire la table de vérité ayant A, B et C en entrée, a, b, …, g en sortie.
2. En déduire les équations logiques simplifiées des sorties.

Exercice 3 **(2 points)**

Voici deux nombres A et B dont on donne le codage en binaire naturel sur 8 bits :

$$A = (01000110)_2$$

$$B = (00101000)_2$$

1. Quelle est la valeur décimale de ces nombres ?
2. Quel est le complément à deux de ces nombres ?
3. Comment vérifier que le résultat est bon ?

S21 - Architecture I

Examen de session 2 - Licence SI - Année 1

28 juin 2018

- Tous les documents, calculatrices et appareils de communication sont interdits -
Le barême est donné à titre indicatif. Durée : 2 h.

PARTIE 1

Cette partie est notée sur 10 points, temps indicatif : 60 mn.

EXERCICE 1. **Architecture** (3 pts) Un constructeur dispose d'un mini ordinateur de démonstration *Binocle* qui fonctionne exclusivement dans le système de numération binaire (en base deux). Binocle possède 64 mots mémoires en mémoire centrale. Ce constructeur veut construire une machine de capacité identique *Octopussy* qui fonctionnera exclusivement dans le système de numération octal (en base 8).

(1) Calculer le nombre N_b de chiffres binaires nécessaires pour représenter 64 objets distincts ? En déduire la taille en bits du compteur ordinal de Binocle.

(2) Calculer le nombre N_o de chiffres octals nécessaires pour représenter 64 objets distincts ? Quelle sera la taille (en octal) de l'adresse d'un mot mémoire pour Octopussy ?

(3) Une instruction de Binocle, ou d'Octopussy, est codée par `code opération + adresse instruction` où le code opération est un chiffre satisfaisant : $0 \leq$ code opération $\leq (7)_{10}$. Quelle est la taille (en bits) d'une instruction pour Binocle ? Quelle est la taille (en chiffres octals) d'une instruction pour Octopussy ?

EXERCICE 2. **Ordinapoche** (4 pts)

On rappelle les codes opérations d'Ordinapoche :

Code	Assembleur	Signification
0	INP	lecture depuis un périphérique d'entrée
1	OUT	affichage sur un périphérique de sortie
2	CLA	mise à zéro d'ACC et addition
3	STO	stocke le contenu d'ACC à l'adresse fournie
4	ADD	addition
5	SUB	soustraction
6	SHT	décalage gauche puis droite de l'ACC
7	JMP	branchement inconditionnel à l'adresse fournie
8	TAC	si ACC $\neq 0$ alors branchement conditionnel à l'adresse fournie
9	HRS	fin de programme

Soit le programme suivant :

Adr.	Instruction	Code	Commentaires
00	INP a	050	lecture du dividende a
01	CLA a	250	
02	STO t	352	
03	INP b	051	lecture du diviseur b
04	CLA a	250	
05	SUB b	551	
06	STO a	350	
07	TAC 10	810	
08	OUT vrai	155	b divise a
09	HRS	900	
10	CLA t	252	
11	SUB un	553	
12	STO t	352	
13	TAC 04	804	
14	OUT faux	154	b ne divise pas a
15	HRS	900	
⋮			
50			a : le dividende
51			b : le diviseur
52			t
53		001	un : le nombre 1
54		000	faux : la valeur "faux"
55		001	vrai : la valeur "vrai"

Ce programme indique si le nombre b divise ou non le nombre a, a et b étant des entiers naturels (sur lesquels on ne fait aucune autre hypothèse).

(1) Faire une table des valeurs, à partir de l'instruction d'adresse 04, comportant les entrées CO (compteur ordinal), RI (registre d'instruction) et ACC (accumulateur) pour les cas suivants : $a = 5$ et $b = 2$; $a = 6$ et $b = 2$.

(2) Quel rôle précis joue le mot-mémoire d'adresse 52 ?

(3) Modifier le programme de tel sorte qu'il affiche le quotient si celui-ci existe et 0 sinon. Pour cela, ajouter **un** bloc d'instructions ainsi qu'une variable et modifier l'une des instructions `TAC` et `OUT`.

EXERCICE 3. **Base de numération et codage** (3 pts)

(1) Changement de base : un nombre s'écrit 753 en base 8. Comment s'écrit-il en base 16 ?

(2) On dispose d'une machine travaillant sur des nombres binaires de longueur 8 (8 bits) en complément à deux. Faire manuellement ce que l'additionneur de la machine ferait automatiquement pour les opérations suivantes : $99 + 35$ et $-61 - 44$. Donner les résultats obtenus en binaire et, en cas d'erreur, indiquer pourquoi.

Partie 2

Cette partie contient 3 exercices et est notée sur 10 points. Temps indicatif : 60 mn

Tous les documents, calculatrices et appareils de communication sont strictement interdits.

Exercice 1 **(4 points)**

On veut réaliser un circuit qui réalise le transcodage d'un nombre A codé en binaire naturel sur 4 bits (A_0 est le LSB de A) en un autre nombre B codé selon le codage Gray.

Questions :

1. Faire la table de vérité.
2. Faire les différents tableaux de Karnaugh.
3. En déduire les équations logiques simplifiées donnant les B_i en fonction des A_k. Utiliser, dès que possible, l'opérateur OU-exclusif !
4. Faire le schéma logique correspondant (un schéma propre est exigé !).

Exercice 2 **(4 points)** : On veut réaliser un additionneur de deux nombres A et B codés en binaire naturel sur 2 bits : on note $D = A + B$ (A_0 est le LSB de A, ...).

Questions :

1. Combien de bits sont nécessaires pour coder D ? Pourquoi ?
2. Faire la table de vérité ayant A et B en entrée et D en sortie.
3. En déduire les équations logiques simplifiées des sorties. On écrira les sorties D_i en fonction des termes $A_i.B_i$ et $A_i \oplus B_i$, et on montrera, notamment, que
$$D_1 = (A_0B_0) \oplus (A_1 \oplus B_1).$$
4. Faire le schéma logique correspondant (un schéma propre est exigé !).

Exercice 3 **(2 points)**

Voici des nombres dont on donne le codage en binaire en complément à deux sur 8 bits :

$$A = (01010110)_2$$
$$B = (10011100)_2$$
$$C = (11100010)_2$$

Questions :

1. Quelle est la valeur décimale de chacun de ces nombres ?
2. Faire en binaire le calcul de $B + C$. Y-a-t-il débordement ? Comment le vérifier ?
3. Faire la même chose pour $B + A$.
4. Vérifier que le résultat décimal obtenu est le bon.