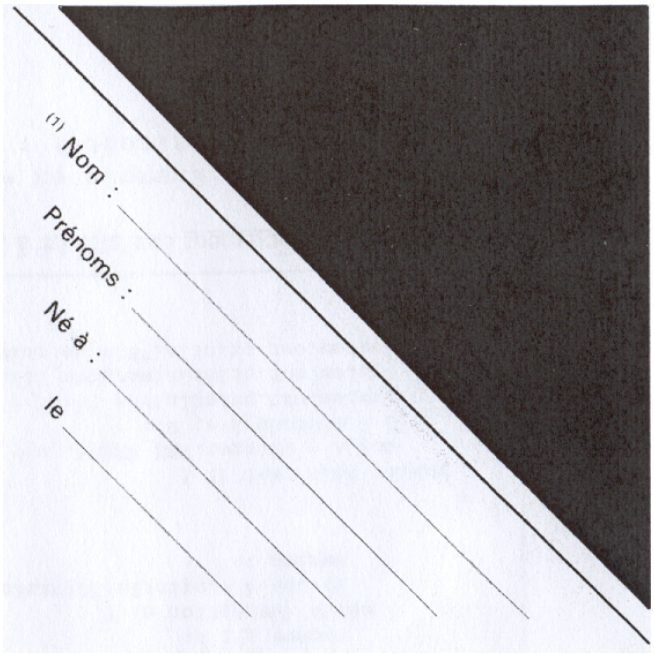
Université de Nice Sophia-Antipolis
L1-INFO
Programmation en Python
Janvier 2015

Durée : 1h30



Seul document autorisé : la feuille de formulaire des primitives Python utilisées.

Note :

A. Chaînes de Caractères

Q1. Qu'appelle-t-on *Unicode* ? Quelles sont les deux fonctions principales de Python permettant d'utiliser *Unicode* ?

```
UNICODE est

                cc1
```

```
Les deux fonctions essentielles
sont
                cc1
```

Q2. Le numéro *Unicode* du caractère € est 8364 en décimal. Comment définiriez-vous au toplevel de Python une variable `prix` dont la valeur est la chaîne de caractères `'18€'` avec un clavier qui ne contient pas le caractère `'€'` ?

```
>>> prix =          cc1
```

Q3. Intéressons-nous à un cas particulier du *code de César* étudié en cours/TP, et connu sous le nom de `ROT13`. Nous ne traiterons que des chaînes de caractères ne contenant que des lettres majuscules sans accents et des espaces, par exemple `'CODE SECRET'`. Le codage `ROT13` consiste dans la chaîne résultat à coder un caractère `A..Z` par la lettre située 13 positions plus loin dans l'alphabet supposé circulaire comme en cours. Programmez la fonction `codage(s)`.
Exemple : `codage('ZOE ZOE')` → `'MBR MBR'` `# l'espace reste intact!`

```
def codage(s) :

                cc1
```

Q4. En quoi le codage `ROT13` est-il particulièrement intéressant ? *Indication : 13 est la moitié de 26…*

cc1

B. Nombres

Q5. Après avoir importé uniquement ce qui est nécessaire, traduisez en Python chaque phrase.

```
# importations
```

cc1

Soit x la variable dont la valeur est la moitié du sinus de la racine cubique de 5

```
x =
```

cc1

Soit y un nombre pair choisi au hasard dans [3,10]

```
y =
```

cc1

Soit moy la fonction qui au nombre entier n > 0 associe la moyenne des entiers de [1,n]

```
def                                    # on suppose n entier > 0
```

cc1

Q6. Programmez une fonction `nbdiv(n)` prenant un entier `n ≥ 2` (ceci est garanti) et retournant le nombre de tous les diviseurs de `n` dans l'intervalle `[1,n]`. Exemple :

```
nbdiv(10) → 4        # ce sont 1, 2, 5, 10
```

```
def nbdiv(n) :
```

cc1

Q7. Déduire de **Q6** une fonction `est_premier(n)` prenant un entier `n >= 2` et retournant `True` si et seulement si `n` est un nombre premier (`False` sinon).

```
def est_premier(n) :
```
cc1

Q8. Du strict point de vue de l'efficacité, quel est l'inconvénient de la fonction `est_premier` ainsi programmée ? *On ne demande pas de programmer une version ameliorée.*

cc1

Q9. Déduire de **Q7** une définition de la variable `S` dont la valeur est la somme de tous les nombres *non premiers* de l'intervalle `[2,1000]`.

cc1

en une ligne ?

Q10. Faites afficher en Python le nombre de chiffres de l'entier `S` défini en **Q9**.

```
>>>
```
cc1

C. Polynômes

Les questions sont indépendantes. Le module `polycreux` est déjà importé. Les polynômes sont creux, l'addition de deux polynômes se note `add` et la multiplication se note `mul`.

Q11. Quelle est l'écriture mathématique du polynôme `[[2, 5], [-1, 4], [-2, 1]]` ?

cc2

Q12. Définissez en quelques lignes les deux polynômes ci-dessous. On ne développera pas à la main ces polynômes bien entendu, on le fera faire par Python en utilisant `add` et `mul`.

`p1 = (x-1)(x+3)` et `p2 = x^2 + (3x^2-1)^6`

cc2

Q13. Définissez la fonction `valeur(p,x)` retournant la valeur en `x` $\in$ **R** du polynôme `p`.

```
def valeur(p,x) :
```
cc2

D. Fichiers et Listes

Bonjour, je suis gérant d'un parc de voitures d'occasion et je travaille avec deux listes de même longueur `LM` *(liste des Marques) et* `LV` *(liste des nombres de Voitures disponibles pour chaque marque). Par exemple (mais on n'utilisera bien entendu pas ces données pour programmer !) :*

```
LM = ['BMW', 'Renault', 'Audi', 'Toyota', 'Peugeot']
LV = [3, 8, 2, 1, 8]
```

signifient qu'il y a 3 véhicules BMW à la vente, 8 Renault, 2 Audi, etc.

Q14. Définissez la variable `maxi` contenant le maximum de la liste `LV`.

```
maxi =
```

cc2

Q15. Ecrivez un petit programme Python permettant d'afficher la liste des marques de voitures ayant exactement `maxi` véhicules disponibles. On affichera le résultat sous la forme suivante *(avec l'exemple ci-dessus)* : `['Renault', 'Peugeot'] : 8 véhicules`

cc2

Q16. Ecrivez un petit script Python qui construise un nouveau fichier `'voitures.txt'` contenant sur chaque ligne une marque suivie d'une *virgule*, suivi du nombre de voitures disponibles de cette marque.

```
BMW,3
Renault,8
Audi,2
...
```

exam

Q17. Ecrivez un petit script Python qui va lire le fichier `'voitures.txt'` puis afficher uniquement le nombre total de véhicules à la vente. Avec l'exemple ci-dessus, il afficherait au toplevel : `22 véhicules à la vente`

exam