Langages de script (LSI4)

Examen du 20 juin 201[illegible] (session 2)

Durée: 3 heures, sans documents

- Vous devez éteindre et ranger vos téléphones.
- Les programmes sont à faire en PYTHON 3.
- Chaque exercice doit être rédigé sur une copie distincte.

Exercice 1 – *Compréhension*

Donnez le résultat des commandes suivantes : étant donné que le ; représente l'enchaînement de deux commandes et que :

```
l = [1,2,3,4,5,6,7,8,9]
d = {1:'a',2:'b',3:'c'}
s = {1,2,3}
```

1. `l[0]`
2. `l[10]`
3. `l[1:5:2]`
4. `l[-2:-1]`
5. `l[-1:-2]`
6. `s.add(2) ; s`
7. `[(i*i,j+j) for i,j in d.items()]`
8. `{d[a]:a for a in d}`

Exercice 2 – *Évènements historiques*

Le but de cet exercice est de faire un script permettant de gérer une petite "chronique" des évènements historiques .

On donnera la chaîne représentant un évènement[1] :

```
Nom: Édit de Nantes
Type: Loi
Personne: Henri IV
Année: 1598
Religion: Protestantisme
Pays: France
Descriptif: un édit de tolérance par lequel le roi reconnaît la liberté de culte aux protesta
```

Le nom et l'année sont toujours présents, tous les autres attributs et leur ordre sont arbitraires (et peuvent varier selon les évènements). À chaque attribut correspond une valeur représentée par une chaîne de caractères (sauf l'année qui est un entier).

Chaque évènement sera représenté par un dictionnaire PYTHON(pour l'exemple ci-dessus, à la clé `"Personne"` correspondra la valeur `"Henri IV"`). Décidez vous-même si la clef `"Nom"` (et sa valeur) sont présents dans le dictionnaire.

1. Source : Wikipedia.

1. Écri ppelée en entrée l'évènement sous la forme d'ur res et r orrespondant à l'évènement.
2. Une hier re ments, séparés par des lignes blanches. La c sentée es clefs sont les noms des évènements et le ictionn s évènements. Écri ppelée en entrée le nof de fichier contenant une yant le la chronique.
3. Écri ppelée t en entrée un dictionnaire et le nom de f n veut ormat décrit à la question précédente.
4. Écri ste_ch n dictionnaire chronique et renvoyant la li ses évè nologique (c-à-d dans l'ordre croissant des
5. Écri ste_at dictionnaire chronique et renvoyant la liste s de to oublons). **Ind** n des r mutation des listes ...) est recommandée).
6. Écri pprime_ trée un dictionnaire `chronique` et une cha tion do que tous les évènements concernant le pay ous les vènements sans attribut "Pays" doivent rest onique.
7. Écr ire.py mande avec la spécification suivante :
   - h CHIER s les évènements d'une chronique stoc R.
   - h CHIER tous les évènements d'une chronique s IER.
   - h CHIER e stockée dans le FICHIER, supprimer t oncern er le résultat dans le même fichier.

   Ind ez uti tions 1 — 6 même si vous ne les avez pas

Exercice *ompte*

Les util pe acilement que les dictionnaires :

```
>>> cn
>>> fo 'blu ue', 'blue']:
...
>>> cn
Counte d': 2
```

Plus chéma s sur ces objets :

```
>>> c 1)
>>> d 2)
>>> c
Counte })
>>> c ent les occurrences positives
Counte
>>> c min(c[x], d[x])
Counte })
>>> c max(c[x], d[x])
Counte })
```

On se propose dans cet exer définir les opération ment sur les diction-
naires pour en faciliter les mani .

1. Écrivez pour cela quatre fon s, moins, et, o ument deux diction-
naires pour lesquels les vale s. Elles d er de la même façon
que les opérations sur les co er le dict dant. Attention, les
dictionnaires ne contiennen ment les mêmes re comme pour les
compteurs dans ces cas-là :

```
>>> c = Counter(a=3, b=
>>> d = Counter(e=4, f=
>>> c | d
Counter({'f': 5, 'e': 4,  'b': 1})
```

Rappel : si on essaie d'affich d'une clef qui n onnaire, on obtient
une erreur `KeyError`. Éviden vous devez faire en ive pas.

2. Les fonctions définies à la q rsonnent le défa leur passe en argu-
ment un dictionnaire qui a a que des entiers ur éviter cela, on va
décider de faire les opératio c'est possible et n. Par exemple, on
veut avoir :

```
>>> d = {'a':2, 'b':[]}
>>> e = {'c':'k', 'a':5}
>>> plus(d, e)
{'a':7, 'b':[], 'c':'k'}
```

Pour faciliter les choses, on de façon rudim me des exceptions.
Voici un exemple d'utilisatio

```
>>> d = {'j':[]}
>>> e = {'f':4}
>>> try:
...     d['j'] + e['f']
... except:
...     print('erreur')
...
erreur
```

Réécrivez **uniquement** la fo qu'elle ait l'aide d'exceptions.