Devoir d'informatique 4

D.Malka – MPSI 2016-2017 – Lycée Saint-Exupéry

24.02.2017

Durée de l'épreuve : 1h00.

L'usage de la calculatrice est autorisé.

L'énoncé de ce devoir comporte 2 pages.

— *Si, au cours de l'épreuve, vous repérez ce qui vous semble être une erreur d'énoncé, signalez le sur votre copie et poursuivez votre composition en expliquant les raisons des initiatives que vous êtes amené à prendre.*
— *Il ne faudra pas hésiter à formuler des commentaires. Le barème tiendra compte de ces initiatives ainsi que des qualités de rédaction de la copie.*
— *La numérotation des exercices doit être respectée. Les résultats doivent être systématiquement encadrés.*
— *Les pages doivent être numérotées de la façon suivante : n°page courante/nombre total de pages.*

Données pour tous les exercices.

On rappelle que :

— La méthode `append()` ajoute l'argument en queue de tableau. Appel : `l.append(elt)`.
— La fonction `len()` renvoie un entier égal à la longueur du tableau passé en argument. Appel : `len(L)`.
— La fonction `float()` convertit l'argument en flottant. Appel `float(n)`.
— La fonction `int()` convertit l'argument en un entier. Si l'argument est un flottant, la fonction `int()` renvoie la partie entière de ce nombre. Appel `int(x)`.
— La fonction `range(a,b,p)` génère la liste suivante : $[a, a+p, a+2p, \ldots, a+jp, \ldots, b-p]$. L'appel `range(a,b)` est équivalent à `range(a,b,1)` et l'appel `range(b)` est équivalent à `range(0,b,1)`.
— La fonction `linspace(a,b,N)` génère la liste suivante : $[a, a+p, a+2p, \ldots, a+jp, \ldots, a+(N-1)p]$ avec $p = \frac{b-a}{N}$.
— Slicing : l'expression `l[i:j]` renvoie la sous-liste comprenant les éléments d'indice i inclus à j exclu de la liste `l`.

Exercice 1 – Représentation des nombres en machine

1. Donner la représentation binaire sur 8 bits de l'entier **naturel** 54.
2. Donner la représentation binaire sur 8 bits de l'entier **relatif** -123 suivant la méthode du complément à deux.
3. Donner la valeur du nombre réel x qu'approche le flottant suivant :

   0 10011101101 1111010000000000000000000000000000000000000000000000

   On rappelle que l'exposant stocké est entier naturel décalé de $+1023$ par rapport à l'exposant codé.
4. On élève itérativement la valeur précédente de x au carré à l'aide d'une calculatrice. Expliquer ce qui se produit à partir d'une certaine itération i_0 et, en supposant que la calculatrice est équipée d'un processeur 64 bits, donner i_0.

Exercice 2 – Trouve le nombre !

On souhaite programmer le jeu qui consiste à deviner en le moins de tentative possible un nombre entier n aléatoire compris entre 0 et N. A chaque proposition, la fonction indique au joueur si la réponse est trop grande ou trop petite. On vous demande de simuler le jeu de l'ordinateur selon le mode :

— *débile* : la fonction `debile(N)` recherche séquentiellement la valeur de n en partant de 0 et incrémentant la proposition de 1 à la réponse suivante, quoi qu'il arrive, jusqu'à trouver.

— *difficile* : la fonction `difficile(N)` qui trouve la réponse en au plus $\log_2(N)$ coups.

Les deux fonctions renvoient la valeur de n correcte ainsi que le nombre de coup joués pour trouver n.

On supposera qu'il existe une fonction `reponse_ordinateur` prenant en argument la valeur r proposée et renvoyant 0 si $r < n$, 1 si $r > n$ et 2 si $r = n$. Les fonctions `debile` et `difficile` appelleront `reponse_ordinateur`.

1. Ecrire la fonction `debile` en langage Python.
2. Ecrire la fonction `difficile` en langage Python.
3. Montrer que le nombre de coup joués par la fonction `difficile` vaut bien au pire $\log_2(N)$.

Exercice 3 – Un programme

On considère le programme suivant :

```python
a=2.0
for n in range(10):
    a=a+0.1
    print(repr(a))
```

où la fonction `repr(a)` permet d'imprimer la valeur exacte de `a` en mémoire.

1. Si on calculait avec des nombres décimaux exactes, quel serait le résultat de ce programme ?
2. Le résultat du programme est donné fig.1.

```
Program Output:
2.1
2.2
2.3000000000000003
2.4000000000000004
2.5000000000000004
2.6000000000000005
2.7000000000000006
2.8000000000000007
2.900000000000001
3.000000000000001
```

Figure 1 – Résultat du programme

   Que constate-t-on ?
3. La représentation binaire exacte de la mantisse de $0,1$ est en fait infinie et périodique :

   100110011001100110011001100110011001100110011001100110011001100........

   Expliquer alors le résultat du programme.