Devoir Surveillé d'Informatique

PTSI B Lycée Eiffel

27 novembre 2020

Exercice 1

Les questions de cet exercice sont indépendantes.

1. Citer un composant électronique dont la découverte a permis de réduire sensiblement la taille des ordinateurs. Quand sont apparus les premiers ordinateurs « à taille humaine » (on n'attend pas une date très précise ici) ?
2. Un nombre entier est codé sur 8 bits par le code 10110101. De quel entier s'agit-il en codage binaire naturel ? De quel entier relatif s'agit-il si on considère un codage en complément à 2 ?
3. Donner le codage sur 32 bits (un bit de signe, 8 bits d'exposant et 23 bits de mantisse) du nombre flottant $\frac{1}{8} + \frac{1}{256} + \frac{1}{1\ 024} = \frac{1}{2^3} + \frac{1}{2^8} + \frac{1}{2^{10}}$.
4. Que signifie le signe USB ? Expliquer brièvement le principe des branchements USB.
5. Expliquer brièvement ce que signifient les caractéristiques suivantes du langage de programmation Python : libre, de bas niveau, multi-plateformes.

Exercice 2

Un nombre entier naturel est dit **polydivisible** si son écriture décimale contient p chiffres et que, pour tout entier $k \leqslant p$, le nombre formé des k premiers chiffres de l'écriture décimale de n est divisible par k.

Ainsi, le nombre 1 624 est polydivisible : c'est un nombre à quatre chiffres, le nombre 16 qui est constitué de ses deux premiers chiffres est divisible par 2, le nombre 162 qui est constitué de ses trois premiers chiffres est divisible par 3 et le nombre 1 624 est lui-même divisible par 4.

Un nombre à trois chiffres abc est donc polydivisible si et seulement s'il est divisible par 3 et si ab est un entier pair.

1. Quelle commande Python permet de déterminer facilement si un nombre entier est pair ?
2. On propose le programme suivant, où le paramètre n est supposé être un entier à trois chiffres :

```
def miaou(n) :
    a=n%10
```

```
b=n-a
return b/10
```

Si on fait tourner cette fonction pour $n = 289$, quelles valeurs vont prendre les différentes variables et quelle sera la valeur retournée par la fonction ?

3. Plus généralement, que calcule le programme précédent ? Aurait-on pu effectuer le même calcul de façon plus rapide ?
4. Écrire une fonction Python **polydiv(n)** qui prend comme argument un entier n dont on suppose qu'il s'agira nécessairement d'un entier à trois chiffres, et qui renvoie True ou False selon que n est polydivisible ou non (on pourra bien sûr réutiliser le programme miaou donné ci-dessus).
5. Écrire de même une fonction Python **polydivbis(n)** qui effectue la même chose qu'à la question précédente, mais cette fois-ci pour un entier n à quatre chiffres.
6. Écrire une suite d'instructions Python permettant de compter le nombre total de nombres entiers à 4 chiffres qui sont polydivisibles (on pourra utiliser la fonction polydivbis de la question précédente même si on n'en a pas écrit une version correcte).

Exercice 3

Si x et y sont deux nombres réels quelconques, on définit une fonction prenant comme variables x et y de la façon suivante :

- $f(x, y) = x^2 + y - 3$ si $x > y$
- $f(x, y) = x^2 - y^2 + 12$ si $x < y$
- $f(x, y) = x - 1$ si $x = y$

1. Écrire une définition de fonction Python prenant comme paramètres x et y et calculant $f(x, y)$.
2. On définit une suite (u_n) de la façon suivante : $u_0 = 0$ et, pour tout entier naturel n, $u_{n+1} = f(u_n^2, u_n)$.
   Écrire une fonction ou une suite d'instructions Python permettant le calcul de u_n pour un entier naturel n quelconque.
3. Écrire une fonction ou une suite d'instructions Python permettant d'afficher la valeur maximale parmi celles des nombres u_0, u_1, …, u_n pour un entier naturel n quelconque. Un bonus sera accordé à ceux qui tenteront d'optimiser le nombre de calculs effectués, ainsi qu'à ceux qui feront afficher en plus de la valeur maximale l'indice du terme de la suite correspondant à cette valeur maximale.