

IPT DS1

Lundi 6 Novembre – Durée 1h

Les bases de l'informatique : melting-pot

1. Représentations des entiers et flottants

1) Donner les représentations des nombres suivants en base 10 :

- a. $(2B4)_{16}$
 b. $(11010010)_2$

2) Effectuer l'addition d'entiers naturels sur 8 bits suivants : $(10010111)_2$ et $(01011001)_2$.

3) On rappelle qu'en représentation en complément à 2 sur n bits (complément à 2^n pour être exact !), on peut représenter tous les nombres entiers de l'intervalle $\llbracket -2^{n-1}, 2^{n-1} - 1 \rrbracket$: si un nombre x vérifie $0 \leq x < 2^{n-1}$, on le code normalement en binaire, sinon on le code par la représentation binaire de $x + 2^n$.

- a. Coder sur 8 bits, en complément à deux, les nombre $x=71$ et $y=-96$
 b. Donner la valeur des entiers relatifs suivants, codés sur 8 bits en complément à deux :

- $M = (01101010)_2$
 • $N = (11000101)_2$

4) Dans cet exercice, on considérera des nombres flottants, donnés par leur représentation par signe, exposant et mantisse. On rappelle ci-dessous le nombre de bits utilisés dans les formats classiques (simple précision 32 bits et double précision 64 bits), et on donne également une représentation de flottants sur 9 bits « maison » que l'on utilisera ici (ça fait moins de chiffres !) Dans ces représentations, l'exposant décalé est un entier naturel codé en binaire, et on utilise un décalage donné par $E_{\max} = 2^{\text{taille de l'exposant décalé}} - 1$. La mantisse se calcule en puissances négatives de 2 de manière standard.

format	signe	exposant décalé	décalage	mantisse	signification (nombre normalisé)
32 bits	1 bit	8 bits	$2^{8-1} - 1 = 127$	23 bits	$(-1)^{\text{signe}} \times 1, \underbrace{\dots}_{\text{mantisse}} \times 2^{\text{exposant décalé}-127}$
64 bits	1 bit	11 bits	$2^{11-1} - 1 = 1023$	52 bits	$(-1)^{\text{signe}} \times 1, \underbrace{\dots}_{\text{mantisse}} \times 2^{\text{exposant décalé}-1023}$
9 bits	1 bit	4 bits	$2^{4-1} - 1 = 7$	4 bits	$(-1)^{\text{signe}} \times 1, \underbrace{\dots}_{\text{mantisse}} \times 2^{\text{exposant décalé}-7}$

On considère la représentation sur 9 bits donnée plus haut.

- a. Donner les valeurs minimales et maximales du facteur $2^{\text{exposant décalé}-7}$.
 b. À quoi sont égaux les nombres suivants ?

- $A=011101111$
 • $B=101111010$

2. Algorithmes et programmation

1) Que renvoie la fonction suivante, prenant en entrée un entier $n \geq 1$:

```
def f(n):
    p=1
    for i in range(1,n+1):
        p=p*i
    return(p)
```

2) On suppose deux variables *masse* et *vitesse* déjà affectées. L'énergie cinétique d'un véhicule est donnée par $energie = \frac{1}{2}masse \times vitesse^2$. Proposer un code qui calcule l'énergie et la stocke dans une variable *energie*, et qui indique par un message à l'écran si le véhicule a une grande énergie cinétique ($energie \geq 1\ 000\ 000$), une énergie cinétique modérée ($100\ 000 < energie < 1\ 000\ 000$) ou faible ($energie \leq 100\ 000$).

3) À l'aide de boucle(s) *for*, calculer les sommes suivantes :

a. La somme :



$$\sum_{i=1}^{10} i^2$$

Par ailleurs, que vaut cette somme d'après le cours de maths ?

b. La somme double :



$$\sum_{i=0}^{100} \sum_{j=1}^{100} ij$$

4) On se donne une liste L et un élément x. Écrire un code permettant de stocker dans la variable *est_present* un booléen indiquant si x est présent dans L.



5) (Questions liées).

a. Sans utiliser la fonction `max(L)`, écrire une fonction *RechMax(L)* prenant en argument une liste L de nombres (flottants) et retournant le maximum de la liste. On parcourra la liste à l'aide d'une boucle *for*.



b. (dur) En utilisant toujours une seule boucle, écrire une fonction *Rech2Max(L)* prenant en argument une liste L de nombres et retournant les 2 plus grands éléments, donnés par ordre décroissant, et sous forme d'une liste. (Exemple : pour [1,3,5,-7], le programme retournera [5,3] comme résultat).



6) La fonction `randint` importée du module `random`, fournit des entiers aléatoires. Ainsi, `random.randint(0,1)` renvoie un entier qui est soit 0 soit 1. Écrire une fonction *temps_1()* sans argument, effectuant des tirages aléatoires, et renvoyant le nombre d'essais qui ont été effectués jusqu'à l'obtention d'un 1.

