université PARIS DIDEROT PARIS 7

EXAMEN PF1 22/10/2014 10^{45}–12^{15}

Aucun document ou support autre que le sujet ou les copies d'examen n'est autorisé.
(la copie ou les brouillons du voisin ne sont pas des supports autorisés).
Éteignez impérativement vos mobiles.

Lorsque des calculs sont nécessaires, il est impératif de les présenter sur la feuille d'examen. Il est aussi nécessaire de **justifier** ses réponses.

1 Exercice (4 points)

1. Sans les convertir, indiquer parmi les nombres suivants, lesquels dans leur écriture en base 2 se terminent par le chiffre 0 ou le chiffre 1 : $(234127)_{10}$, $(127842344234)_{10}$, $(1000000000001)_{10}$, $(100000010)_{10}$.
   - $(234127)_{10}$ est impair, son écriture en binaire terminera donc par le chiffre 1,
   - $(127842344234)_{10}$ est pair, son écriture en binaire se terminera donc par le chiffre 0,
   - $(1000000000001)_{10}$ est impair, son écriture en binaire se terminera donc par le chiffre 1,
   - $(100000010)_{10}$ est pair, son écriture en binaire se terminera donc par le chiffre 0.r
2. Convertir en base 2, le nombre $(443)_{10}$.
   - par divisions entières successives on obtient 443/2 quotient 221 reste 1. 221/2 quotient 110 reste 1. 110/2 quotient 55 reste 0. 55/2 quotient 27 reste 1. 27/2 quotient 13 reste 1. 13/2 quotient 6 reste 1. 6/2 quotient 3 reste 0. 3/2 quotient 1 reste 1. On lit les restes à l'envers (ils correspondent aux plus grandes puissances de 2) et on obtient $(110111011)_2$
3. Sans effectuer le calcul, comment s'écrit en base 2 le résultat du calcul (en base 10) suivant : $4 \times 443 + 1$?
   - On sait que $(443)_{10}$ s'écrit $(110111011)_2$ mais peut importe son écriture de 443 en base 2, disons qu'il s'agisse du mot w. 4 en base 2 s'écrit 100 et correspond à un décalage de 2 positions vers la gauche puisqu'on multiplie par le carré de la base. Donc on obtient $w00$. Si ensuite on ajoute 1 on obtient nécessairement $w01$.
4. Calculer en base 2 la somme de 101100011101011 et de 101111010001011.
   - On pose l'addition et on fait apparaître les retenues (ici en haut) :

```
  101100010001011
   101100011101011
+ 101111010001011
------------------
  1011011101110110
```

5. Calculer en base 2 le produit de 101100011101011 par 10111.
   - On pose la multiplication, on fait les additions en faisant apparaître les retenues (ici après) :

```
        101100011101011
      ×           10111
-----------------------
        101100011101011
+     101100011101011.
+    101100011101011..
+ 101100011101011....
                 11110
               10
               1
            11
            1
          10
  001111110
-----------------------
   111111110100011101
```

2 Exercice (4 points)

Dans cet exercice on s'intéresse au système de numération positionnelle utilisant les 4 chiffres `O`, `I`, `A` et `E` représentant respectivement aucun élément, une unité, deux unités et trois unités.

1. À quelle base de numération ce système correspond-il ?
   - Il y a quatre symboles représentant des nombres différents, c'est donc la base 4
2. Quelles sont les tables d'addition et multiplication de ce système ?

— la table d'addition :

+	O	I	A	E
O	O	I	A	E
I	I	A	E	IO
A	A	E	IO	II
E	E	IO	II	IA

La table de multiplication :

×	O	I	A	E
O	O	O	O	O
I	O	I	A	E
A	O	A	IO	IA
E	O	E	IA	AI

3. À quel nombre de la base 10 le nombre `AIE` correspond-il ?
   — `A`=2, `I`=1, `E`=3, donc `AIE` c'est $(213)_4$ donc $2 \times 4^2 + 2 \times 2^1 + 3 \times 2^0$=32+4+3=39
4. L'addition suivante est-elle correcte : `IA+IA=EO` ?
   — Posons l'addition (avec les retenues au-dessus) :

```
  OI
   IA
 + IA
 ----
  EO
```

3 Exercice (8 points)

Dans cet exercice on s'intéresse à la représentation des nombres en base 2 sur 6 bits.

1. Combien de nombres peut-on représenter au plus ?
   — il y a 2^6 mots différents de 6 bits, soit 64. On peut donc représenter 64 nombres différents.
2. Si la representation est non signée, quels sont les nombres entiers représentés ? Donner votre réponse sous la forme d'un intervalle.
   — cela signifie que l'on souhaite représenter uniquement des nombres positifs (ou nul) soit les nombres entiers de l'intervalle $[0, 64[$ ou $[0, 63]$.
3. Dans la représentation non signée, quels sont les codages des nombres $(23)_{10}$ et $(62)_{10}$?
   — par divisions par 2 on obtient 23/2 — (11, 1), 11/2 — (5, 1), 5 — (2, 1), 2/2 — (1, 0) (en notant (q, r)). Donc $(23)_{10} = (10111)_2$, mais l'écriture sur 6 bits est 010111.
   — par divisions par 2 on obtient 62/2 — (31, 0), 31/2 — (15, 1), 15/2 — (7, 1), 7/2 — (3, 1), 3/2 – (1, 1). Donc $(62)_{10} = (111110)_2$, mais l'écriture sur 6 bits est 010111. (On sait aussi que 63 s'écrit 111111 donc 62 s'écrit 111110).
4. Dans la représentation non signée, le mot `111000` correspond-il à un nombre divisible par 8 ?
   — Oui. 8 est une puissance de 2, $8=2^3$. Donc un multiple de 8 écrit en binaire terminera nécessairement par une suite de trois 0 ; ce qui est le cas.
5. Donner deux instructions Java différentes permettant d'obtenir le quotient de la division entière d'un nombre contenu dans une variable de type `int` et de nom `n` par le nombre 16.
   — `n/16` (le signe `/` est celui de la division entière si les arguments sont de type `int`),
   — `n>>4`, car 16 est une puissance de 2, $16=2^4$, diviser par 16 c'est donc décaler de 4 bits vers la droite.
6. Si la représentation est signée en complément à 2, quels sont les nombres relatifs représentés ? Donnez l'intervalle.
   — L'intervalle sera $[-32, 32[$, ou $[-32, 31]$.
7. Dans la représentation signée en complément à deux, quels sont les codages des nombres $(24)_{10}$ et $(-23)_{10}$?
   — 23 s'écrit 010111 (sur 6 bits) donc 24 s'écrit 011000.
   — pour écrire -23 on peut inverser les chiffres de l'écriture de 23, soit 101000 et ajouter 1 donc 101001. On peut aussi calculer 64-23 soit $1000000 - 010111$ (en posant le calcul on obtient le résultat).
8. Dans la représentation signée en complément à deux, à quel nombre en base 10 correspond le codage `111110` ?
   — Le nombre est négatif puisque son bit de poids fort est 1. Pour calculer sa valeur absolue, on peut inverser les chiffres pour obtenir 000001 puis ajouter 1 ce qui donne 000010 soit 2 en base 10. On peut aussi remarquer que 111111 est le nombre -1 en base 10, le nombre qui précède est 111110 donc -2.
9. Effectuer dans la représentation signée en complément à deux les opérations suivantes (en précisant si le résultat peut être considéré comme correct dans l'arithmétique ordinaire) 011000+010011, 001111+010000, 010111-001011, 000111×000111.

— $011000 + 010011 = 101011$, en complément a deux on obtient donc un nombre négatif en additionnant deux nombres positifs. On peut considérer ce calcul comme faux dans l'arithmétique ordinaire,

— $001111 + 010000 = 011111$, aucun problème, l'addition de deux nombres positifs donne un nombre positif. Ce calcul correspond au calcul obtenu dans l'arithmétique ordinaire,

— $010111 - 001011 = 001100$, aucun problème non plus,

— $000111 \times 000111 = 110001$, ne correspond pas à un opération correcte dans l'arithmétique ordinaire puisque la multiplication de deux nombres positifs donne un nombre positif (le résultat obtenu dans la complémentation à deux est négatif).

4 Exercice (4 points)

Soit le bout de programme Java suivant :

```
1 int i = 10;
2 short s = 32;
3 byte b = 4;

5 int résultat = i+s-b;
6 System.out.println(résultat);

8 byte résultat2 = (byte)(résultat*résultat);
9 System.out.println(résultat2);
```

Qu'affichera t-il lors de son exécution ?

— Le résultat obtenu sera

```
38
-92
```

La valeur de `résultat` est obtenue par conversion de chaque opérande dans le type `int`, on obtient donc 10+32-4 soit 28. Les calculs intermédiaires ne provoquent aucune erreur.

La valeur de `résultat2` est obtenue d'abord par calcul dans le type `int` de 38×38, soit 1444 qui s'écrit sur 32 bits comme 00000000000000000000010110100100. Quelque soit la valeur de la multiplication, la machine ne peut garantir que le résultat puisse être rangé dans la variable de type `byte` (8 bits), par conséquent on a été obligé de forcer la conversion vers le type `byte`, ce qui provoque le stockage des seuls 8 derniers bits soit 10100100 ce qui est la présentation en complément à deux du nombre -92.