Principes de fonctionnement des machines binaires

TP1 – Représentation des nombres en machine

Exercice 1: Utiliser le programme bc dans une fenêtre Shell pour faire des calculs dans les bases que vous voulez... Veri er les calculs faits dans les feuilles de TD precedentes.

Pour les exercices suivants, nous allons travailler avec le langage JAVA. Pour cela, il faut utiliser un editeur de texte pour ecrire des programmes, et une fenêtre Shell dans laquelle on donnera les instructions de compilation et d'execution des programmes...

Exercice 2: On veut travailler sur le codage des entiers sur un octet (8 bits).

1. Ecrire une procedure `ConversionBinaire(int N)` qui prend en argument un entier `N`. Cette procedure devra d'abord veri er que `N` est une valeur possible pour un entier non signe sur un octet, puis elle calculera les valeurs des huit bits `c0`, `c1`, `c2`, `c3`, `c4`, `c5`, `c6`, et `c7`.
2. Ecrire une procedure `ConversionDecimale(int c7, int c6, int c5, int c4, int c3, int c2, int c1, int c0)` qui prend en argument un codage binaire. Cette procedure calculera la valeur representee par ce codage pour des entiers non signes (on supposera que les parametres sont des 0 ou 1).
3. Reprendre les deux questions precedentes avec des entiers signes. Ecrire les procedures `ConversionBinaireSigne(int N)` et `ConversionDecimaleSigne(int c7, int c6, int c5, int c4, int c3, int c2, int c1, int c0)`

Exercice 3: Ecrire une procedure `ConversionBinaire2(long N)` qui prend en argument un entier `N` (non signe) en deduire un codage binaire que l'on a chera au et a mesure du calcul (on ne se limite plus a un octet !). On commencer l'a chage par les bits de poids faibles.

Exercice 4: En Java, on dispose de trois types d'entiers signes : `byte`, `short`, `int` et `long`. Par chacun, trouver le plus petit entier et le plus grand entier que l'on peut manipuler. Justi er vos reponses avec un exemple.

En deduire un programme base sur une boucle for qui ne marche pas en raison d'un probleme de codage des entiers (c'-a-d. un type non adapte a la grandeur des valeurs utilisees).

Exercice 5: Essayer le code Java suivant :

```
int n=0x12345678;
float f=n;
int m = (int) f;
System.out.println(n);
System.out.println(f);
System.out.println(m);
```

Pourquoi `n` et `m` ont elles des valeurs di erentes ?

Exercice 6:

```
for (float r=0; r!= 20; r=r+0.5f)
   System.out.println(r);
```

Que se passe-t-il ? Remplacer le pas `0.5f` par `0.1f`, expliquer... Et par `1/3f` (essayer aussi avec `1/3`) ?