

## Partiel de l'UE INF f1 - Durée 1h30 Documents autorisés

### Exercice 1 (7 points, environ 30 minutes)

a) Écrire  $(10)_{16}$  en base 2 puis en base 8.

$(10000)_2$  en écrivant chaque chiffre sur 4 bits et  $(20)_8$  en groupant par paquets de 3 les chiffres binaires.

b) Écrire  $(1001,01)_2$  en base 10 et en hexadécimal.

$9,25 (1 \times 8 + 1 \times 1 + 1 \times 1/4)$  et  $(9,4)_{16}$  en groupant par paquets de 4 les chiffres binaires.

c) Écrire  $(-8)_{16}$  sur un octet en utilisant la méthode du complément vrai.

$8 = 00001000$  (ne pas oublier d'écrire les 8 chiffres) et donc  $-8 = 11111000$

d) Écrire  $-0,5$  sur 16 bits avec 6 bits d'exposant.

$-0,5 = (-0,1)_2 = -1,0 \times 2^{-1}$

$-1$  est représenté par  $-1+2^5 -1 = 30$ , donc **011110**. La résultat est donc **1011110000000000**

e) En Java, un *float* est codé sur 4 octets, avec 23 chiffres de mantisse. Quel nombre en base 10 est représenté par ce float? **1100000100001000000000000000000000**

Il y a 8 bits pour l'exposant puisqu'il y a 32 bits en tout.

L'exposant est donc 10000010 qui vaut 130 ( $128+2$ ) et il faut lui retirer l'excédent qui vaut  $2^7 = 128-1 = 127$ , ce qui nous donne 3.

Le nombre est donc  $1,000100... \times 2^3 = (1000,1)_2 = 8,5$ .

### Exercice 2 (4 points, environ 20 minutes)

a) Écrire un programme qui lit une chaîne de caractères et qui crée, sans l'afficher, une autre chaîne ne contenant que les 'A' ou les 'B', de cette chaîne, dans le même ordre. Par exemple, si la chaîne lue est "ABRACADABRA", la chaîne résultat sera "ABAAABA".

Le gabarit de parcours de chaîne est :

```
i=0
while (i<chaine.length()) {
    ..... // le caractère courant est chaine.charAt(i)
    i++;
}
```

donc la solution est :

```
Scanner s = new Scanner(System.in);
String chaineDonnee, chaineResultat;
System.out.print("Entrez la chaîne");
chaineDonnee=s.next(); // lecture de la chaîne donnée
chaineResultat=""; // initialement, la chaîne résultat est vide
int i=0; // index dans la chaîne
while (i<chaineDonnee.length()) { // tant qu'on n'est pas au bout de la chaîne donnée
    if (chaineDonnee.charAt(i)=='A' || chaineDonnee.charAt(i)=='B')
        chaineResultat = chaineResultat + chaineDonnee.charAt(i);
    i++; // attention à ne pas placer le i++ dans le if !
}
```

```
s.close();
```

### Exercice 3 (7 points, environ 30 minutes)

Écrire un programme qui lit deux chaînes de même longueur et qui indique la chaîne qui est la première dans l'ordre lexicographique (l'ordre du dictionnaire).

L'opérateur < peut être utilisé pour comparer deux caractères.

Par exemple :

```
Première chaîne : EXAMENS
Seconde chaîne : EXEMPLE
La première est EXAMENS.
```

```
Première chaîne : TRAM
Seconde chaîne : AVION
La première est AVION.
```

```
Scanner s = new Scanner(System.in);
String chaine1,chaine2;
int i=0;
System.out.print("Première chaîne : ");
chaine1=s.next(); // lecture de la chaîne donnée
System.out.print("Seconde chaîne : ");
chaine2=s.next(); // lecture de la chaîne donnée
while (i<chaine1.length() && chaine1.charAt(i)==chaine2.charAt(i))
    i++;
if (i>=chaine1.length()) // attention à ne pas utiliser l'autre condition en premier !
    System.out.println("Les deux chaînes sont identiques.");
else if (chaine1.charAt(i)<chaine2.charAt(i))
    System.out.println("La première est "+chaine1);
else
    System.out.println("La première est "+chaine2);
s.close();
```

### Exercice 4 (2 points, environ 10 minutes)

Réécrire la condition suivante sans utiliser le signe ! :

```
while ( ! ((debut != 1) || ((a > 3) && (a <= 8))))
```

```
while (debut==1 && (a<=3 || a>8))
```