

Examen partiel de l'UE INF f1 – Octobre 2019 – 1 heure 30
Une feuille A4 autorisée. Calculatrice interdite.

Il est inutile d'écrire le début des programmes : import..., ...class..., ...main. L'indentation correcte est prise en compte dans la notation.

Exercice 1 (5 pts = 1+1+2+1, environ 25 minutes)

a) Écrire 20 en base 2 et en base 16.

$$20 = (10010)_2 = (14)_{16}$$

b) Écrire $(10010001111, 10111)_2$ en base 8 et en base 16.

$$(10010001111, 10111)_2 = (2217,56)_8$$

$$(10010001111, 10111000)_2 = (48F, B8)_{16}$$

c) Écrire le flottant 7,7 sur 16 bits avec 7 bits de mantisse.

Les 16 bits se répartissent donc en 1 bit de signe, 8 bits d'exposant, 7 bits de mantisse

$$0,7 \times 2 = 1,4$$

$$0,4 \times 2 = 0,8$$

$$0,8 \times 2 = 1,6$$

$$0,6 \times 2 = 1,2$$

$$0,2 \times 2 = 0,4$$

...

$$\text{donc } 7,7 = (111,10110011001100\dots)_2 = (1,1110110011001100\dots)_2 \times 2^2$$

Le signe est positif donc le bit de signe est 0

$$\text{L'exposant est } 2(00000010) + \text{l'excédent}(01111111) = 10000001$$

La mantisse est 1110110

Les 16 bits sont donc 0 10000001 1110110

d) Pourquoi ne peut-on pas représenter exactement $(1,11011011101)_2$ sur un flottant avec 10 bits de mantisse ? Quel nombre **en base 16** voulait-on représenter et quel nombre représente-t-on en réalité ?

Parce qu'il y a 11 bits après le "1," et qu'on va donc perdre le dernier.

On voulait représenter $(1,110110111010)_2 = (1, DBA)_{16}$ et on représente $(1,110110111000)_2 = (1, DB8)_{16}$

Exercice 2 (3 points, environ 11 minutes)

Écrire un programme qui lit une chaîne et qui affiche l'avant-dernier caractère s'il existe. Si ce caractère n'existe pas, le programme ne doit rien afficher et ne doit pas provoquer d'erreur. Exemple :

Chaîne ? MIASHS

Avant-dernier caractère : H.

```
Scanner s = new Scanner(System.in);
String chaine;
System.out.print("Chaîne ? ");
chaine=s.nextLine();
if (chaine.length()>=2)
    System.out.println("Avant-dernier caractère : "+chaine.charAt(chaine.length()-2)+".");
```

Exercice 3 (4 points, environ 18 minutes)

Écrire un programme qui lit un entier n et qui affiche $n!$. Pour rappel $n!=1\times 2\times 3\times \dots \times n$. Exemple :

Valeur ? 7

7! = 5040

```
Scanner s = new Scanner(System.in);
int i,n, fact=1;
System.out.print("Valeur ? ");
n=s.nextInt();
for(i=2;i<=n;i++)
    fact=fact*i;
System.out.println(n+"! = "+fact);
```

Exercice 4 (4 points, environ 18 minutes)

Écrire un programme qui, étant donné un nombre de secondes inférieur à 10.000, l'écrit sous la forme heures, minutes, secondes. **Ne pas écrire les valeurs nulles.** Exemples :

Nombre de secondes ? 1000

1000 secondes = 6 min 40 s

Nombre de secondes ? 3602

3602 secondes = 1h 2 s

```
Scanner s = new Scanner(System.in);
int sectotal, heures, minutes, secondes;
System.out.print("Nombre de secondes ? ");
sectotal=s.nextInt();
heures = sectotal/(3600);
minutes = (sectotal%3600)/60;
secondes = sectotal%60;
if (heures>0)
    System.out.print(heures+"h ");
if (minutes>0)
    System.out.print(minutes+"min ");
if (secondes>0)
    System.out.print(secondes+"s");
```

Exercice 5 (4 points, environ 18 minutes)

Écrire un programme qui demande 20 nombres à l'utilisateur (utilisez une boucle !) et qui indique à la fin lequel est le plus petit. Exemple :

Nombre 1 ? 7

Nombre 2 ? 3

...

Nombre 20 ? 5

Le plus petit était 3.

```
Scanner s = new Scanner(System.in);
int nb, plusPetit;
System.out.print("Nombre ? ");
plusPetit=s.nextInt();
for(int i=2;i<=20;i++) {
    System.out.print("Nombre ? ");
    nb=s.nextInt();
    if (nb<plusPetit)
        plusPetit=nb;
}
System.out.println("Le plus petit était "+plusPetit+".");
```