

Partiel de l'UE INF f1 – Novembre 2017 - 1h30
Une feuille A4 autorisée

*Il est inutile d'indiquer les premières lignes des programmes Java :
import..., ...class..., ...main... En revanche, les entêtes des actions et fonctions sont
requis.*

Les premières puissances de 2 sont : 256 128 64 32 16 8 4 2 1

Exercice 1 (4 pts, environ 20 minutes)

a) Écrire 4,3 en base 2 avec 3 chiffres après la virgule, puis en base 16.

Partie entière : $4 = (100)_2$

Partie décimale : $0,3 \times 2 = 0,6$. $0,6 \times 2 = 1,2$ $0,2 \times 2 = 0,4$

donc $4,3 = (100,010)_2$

Pour passer en base 16, on fait des groupements de 4 bits car $2^4 = 16$. Donc
 $4,3 = (0100,0100)_2 = (4,4)_{16}$

b) Écrire l'entier -32 sur un octet avec la méthode du complément vrai. L'additionner à
lui-même. Y a-t-il report ? Débordement ?

$32 = (00100000)_2$ donc $-32 = 11100000$

On pose l'addition :

```

  11100000
+  11100000
-----
=  11100000

```

donc il y a un report (9^e bit) mais pas de débordement parce que la 8^e colonne nous
indique qu'on ajoute un négatif (**1**) et un négatif (**1**) et qu'on obtient un négatif (**1**).

c) Que vaut en base 10 ce *float* (4 octets dont 23 chiffres de mantisse) :

11000100 00000000 00000000 00000000

Le signe est 1 donc c'est un négatif

L'exposant est constitué des 8 chiffres suivants (10001000). Cet exposant représente la
valeur 9 ($128 + 8$ auquel on soustrait l'excédent 01111111 qui vaut $2^{8-1} - 1 = 127$, ce qui
donne 9).

La mantisse est constitué des 23 chiffres qui restent (00000000000000000000000)

Donc le nombre est $-1,0000... \times 2^9 = -512$

Exercice 2 (5 points, environ 20 minutes)

Écrire une **action** qui, étant donné une chaîne de caractère et un entier N strictement inférieur à la longueur de la chaîne, affiche un caractère sur N. Par exemple, avec les paramètres "ABCDEFGHJKLMNOPQRS" et 4, l'action affichera un caractère sur 4 (le 1^{er}, le 5^e, le 9^e, le 13^e, etc.) : AEIMQ

```
static void unSurN(String ch,int n) {
    int i=0;
    while (i<ch.length()) {
        System.out.print(ch.charAt(i));
        i+=n;
    }
}
```

Exercice 3 (5 points, environ 20 minutes)

Écrire une **fonction** qui étant donné un tableau de caractères retourne une chaîne composée de la concaténation de ces caractères, dans l'ordre inverse. Par exemple, avec le tableau {'s','h','s','a','i','m'}, la fonction doit retourner la chaîne "miashs".

```
static String affinv (char[] t) {
    String res="";
    int i=0;
    while (i<t.length) {
        res=t[i]+res;
        i++;
    }
    return(res);
}
```

Exercice 4 (6 points, environ 30 minutes)

On veut écrire un programme pour enregistrer les coureurs qui ont bien terminé une course à pied en montagne. Chaque fois qu'un coureur arrive, on saisit son numéro de dossard. A la fin, le programme affiche les coureurs qui ne sont pas arrivés. On utilise pour cela un tableau de booléens, tous initialisés à false. Quand un coureur arrive, on met à true la case dont l'indice correspond à son numéro de dossard. Les numéros de dossard commencent à l'indice 1. Il y a 100 coureurs.

a) Écrire une **action** qui étant donné un tableau de booléens affiche les indices des cases qui contiennent la valeur false (un par ligne).

```
static void afficheNonArrives (boolean[] t) {
    int i=0;
    while (i<t.length) {
        if (t[i]==false)
            System.out.println(i);
        i++;
    }
}
```

b) Écrire un **programme** qui crée un tableau de booléens et qui le remplit à l'aide d'une boucle avec les valeurs false. Puis le programme demande en boucle à l'utilisateur un numéro de dossard, et met à true la case correspondante dans le tableau. Quand l'utilisateur saisit -1 comme numéro de dossard, on sort de la boucle et on appelle l'action qui affiche les numéros de dossard des coureurs non arrivés. Voici un exemple ci-contre.

```
Numéro de dossard : 45
Numéro de dossard : 7
Numéro de dossard : 33
Numéro de dossard : 59
...
Numéro de dossard : 12
Numéro de dossard : -1
Les coureurs non arrivés sont :
17
21
34
35
89
```

```
public static void main(String [] args) {
    Scanner s = new Scanner(System.in);
    boolean []t=new boolean[101];
    int i,dossard;
    for(i=0;i<t.length;i++) // initialisation du tableau
        t[i]=false;
    do {
        System.out.print("Numero de dossard ? ");
        dossard=s.nextInt();
        if ((dossard>=1) && (dossard < t.length))
            t[dossard]=true;
    } while (dossard != -1);
    System.out.println("Les coureurs non arrives sont :");
    afficheNonArrives(t);
}
```