

$$f = 010000\dots$$

[illegible]

```
1:      s          e          f
    0      01111111111      00000000000000000000000000000000000000000000000000000
```

```
2: s e f  
0 100000000000 0000000000000000000000000000000000000000000000000
```

4. Nous pouvons représenter une date avec 3 variables a, m, q qui sont liées respectivement à l'année, le mois et le quantième.  
Les mois sont encodés par le numéro du mois (i.e. 1=janvier, 2=février, etc) et le quantième c'est le jour dans le mois (i.e. 1=premier jour du mois).  
Nous pouvons donc représenter le 16 octobre 2012 avec ces déclarations de variables :

```
var a = 2012;
```

```
var m = 10;
```

```
var q = 16;
```

La déclaration de variable suivante permet de lier la variable j au jour de la semaine correspondant aux valeurs de a, m et q :

```
var j = (Math.floor(23*m/9) + Math.floor((a*12+m-3)/12) + Math.floor((m+9)/12)*4
+ Math.floor(Math.floor((a*12+m-3)/12)/4) -
Math.floor(Math.floor((a*12+m-3)/12)/100) +
Math.floor(Math.floor((a*12+m-3)/12)/400) + q + 5) % 7 + 1;
```

Le jour de la semaine c'est un entier dans l'intervalle 1 à 7. La valeur 1 représente dimanche, 2 représente lundi, etc. Donc dans ce cas particulier j sera liée à 3 puisque le 16 octobre 2012 est un mardi.

La formule utilisée est difficile à lire car elle contient des parties répétées.

Utilisez des déclarations de variables pour décomposer le calcul de  $j$  en plusieurs étapes. Il faut éviter autant que possible de refaire un calcul plus d'une fois.

**Solution :**

```
var a = 2012;
```

```
var m = 10;
```

```
var q = 16;
```

```

var anneeAjustee = Math.floor((a*12+m-3)/12);

/* Calculs intermédiaires pour éviter de se retrouver avec une ligne trop longue */
var u = Math.floor(23*m/9) + anneeAjustee;
var v = Math.floor((m+9)/12)*4 + Math.floor(anneeAjustee/4);
var w = Math.floor(anneeAjustee/100) + Math.floor(anneeAjustee/400);

var j = (u + v - w + q + 5) % 7 + 1;

print(j);

```

5. Voici des expressions qui font des calculs sur des entiers encodés en représentation complément à 2 avec 32 bit.

Quelle est la valeur de chaque expression?

Donnez la séquence de bits correspondants à chacune des opérandes et à la valeur résultante. Vérifiez vos résultats avec codeBoot.

3 & 5  
 3 | 5  
 3 ^ 5  
 3 << 4  
 100 >> 2  
 ~ 12

Sol:

**3 & 5:**

```

3 = 0000 0000 0000 0000 0000 0000 0000 0011
5 = 0000 0000 0000 0000 0000 0000 0000 0101
3&5 = 0000 0000 0000 0000 0000 0000 0000 0001
    = 1

```

**3 | 5:**

```

3|5 = 0000 0000 0000 0000 0000 0000 0000 0111
    = 7

```

**3 ^ 5:**

```
3^5 = 0000 0000 0000 0000 0000 0000 0000 0110
      = 6
```

**3 << 4:**

```
3<<4 = 0000 0000 0000 0000 0000 0000 0011 0000
      = 48
```

**100 >> 2:**

```
100   = 0000 0000 0000 0000 0000 0000 0110 0100
100>>2 = 0000 0000 0000 0000 0000 0000 0001 1001
      = 25
```

**~ 12:**

```
12 = 0000 0000 0000 0000 0000 0000 0000 1100
~12 = 1111 1111 1111 1111 1111 1111 1111 0011
     = 1 + 2 + 16 + 32 + ... + 2**30 - 2**31
     = -13
```