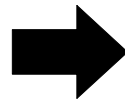
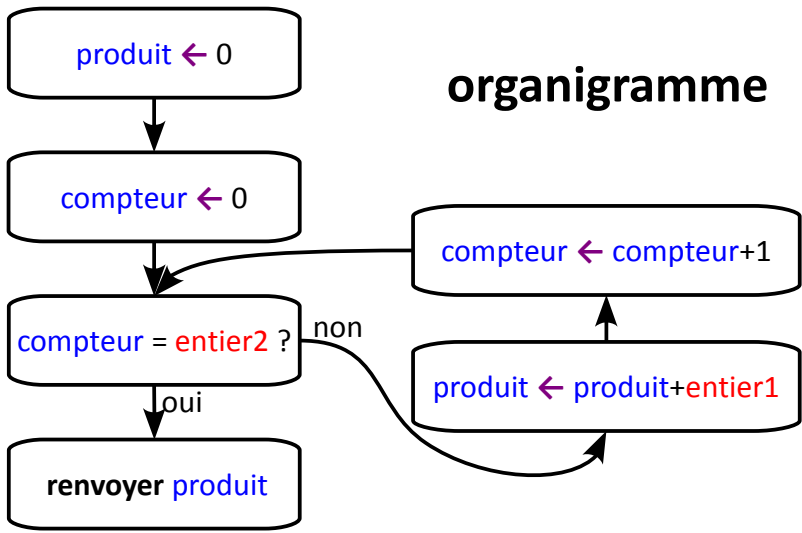


# Dictionnaire pseudo-code / Java

	Pseudo-code	Java
Déclaration d'un algorithme	<b>Addition</b> Entrées : entiers $i$ et $j$ Sortie : somme $i+j$ (entier) Début ... renvoyer ... Fin	public static int <b>Addition</b> (int $i$ , int $j$ ){ ... return ... }
Déclaration d'une variable	Variables : entier $i$	int $i$ ;
Affectation	$i \leftarrow 1$	$i = 1$ ;
Test	Si $i=1$ alors ... Sinon ... FinSi	if ( $i==1$ ){ ... } else { ... }
Boucle	Tant que $i<3$ : ... Fin TantQue	while ( $i<3$ ) { ... }

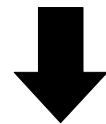
# De l'organigramme au code Java



**Multiplication :**                      **pseudo-code**

**Entrées :** deux entiers *entier1* et *entier2*  
**Sorties :** le *produit* de *entier1* et *entier2*  
**Variables :** entiers *compteur* et *produit*

Début  
    *compteur* ← 0  
    *produit* ← 0  
    Tant que *compteur* < *entier2* faire :  
        |     *produit* ← addition(*produit*, *entier1*)  
        |     *compteur* ← addition(*compteur*, 1)  
    Fin tant que  
    renvoyer *produit*  
Fin



public static int multiplication(int *entier1*, int *entier2*){

```
int compteur=0;  
int produit=0;  
while (compteur<entier2){  
    produit=produit+entier1;  
    compteur=compteur+1;  
}  
return produit;
```

**code Java**

}

# Codage binaire

## La "minute votes SMS"

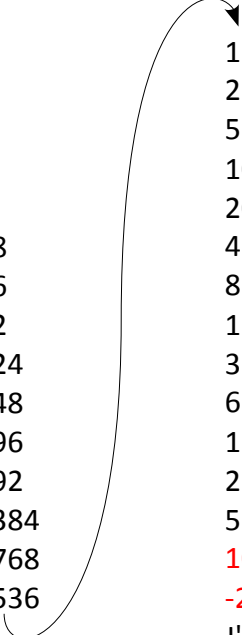
Programme Java :

```
import java.io.*;
public class Boucle{
    public static void main(String [] arg){
        int i=1;
        while(i>0){
            i=i*2;
        }
        System.out.print("J'ai fini !");
    }
}
```

Le programme affiche "J'ai fini" parce que les entiers "int" Java ne sont pas de "vrais entiers" mais des entiers entre -2 147 483 648 et 2 147 483 647

Résultat du programme :

2	
4	131072
8	262144
16	524288
32	1048576
64	2097152
128	4194304
256	8388608
512	16777216
1024	33554432
2048	67108864
4096	134217728
8192	268435456
16384	536870912
32768	1073741824
65536	-2147483648
	J'ai fini!!



## La "minute mathématique"

Pour le stockage comme pour le traitement d'instructions, il est nécessaire que toutes les données traitées par un ordinateur soient codées en **binaire**, par des **0** et des **1**.

Exemple de nombre entier en binaire : 1101100001101

0	0	0	1	1	0	1	1	0	0	0	0	1	1	0	1
$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

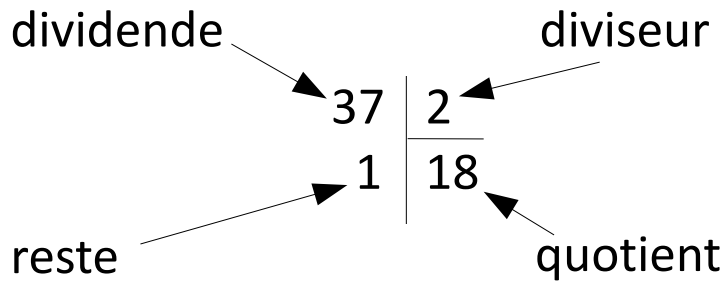
$$2^{12} + 2^{11} + 2^9 + 2^8 + 2^3 + 2^2 + 2^0 = 4096 + 2048 + 512 + 256 + 8 + 4 + 1 = 6925$$

# Codage binaire

Pour le stockage comme pour le traitement d'instructions, il est nécessaire que toutes les données traitées par un ordinateur soient codées en **binaire**, par des **0** et des **1**.

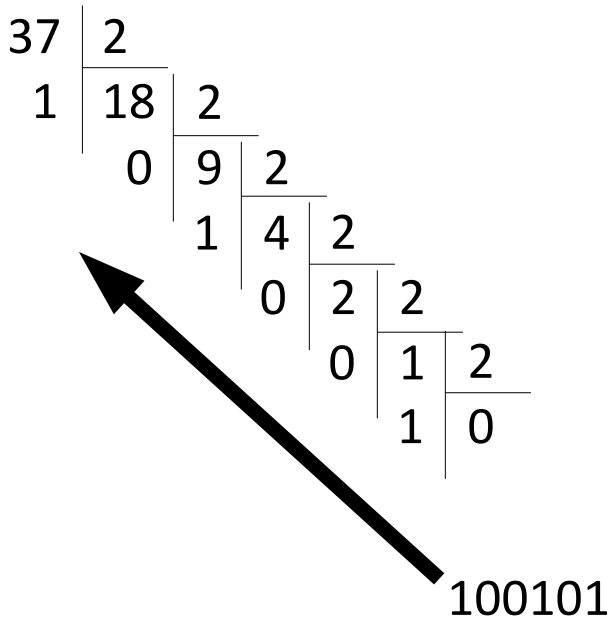
*Exemple : écrire 37 en binaire ?*

### Division euclidienne :



en Java :  $1=37\%2$

en Java :  $18=37/2$



Pour faire son geek :

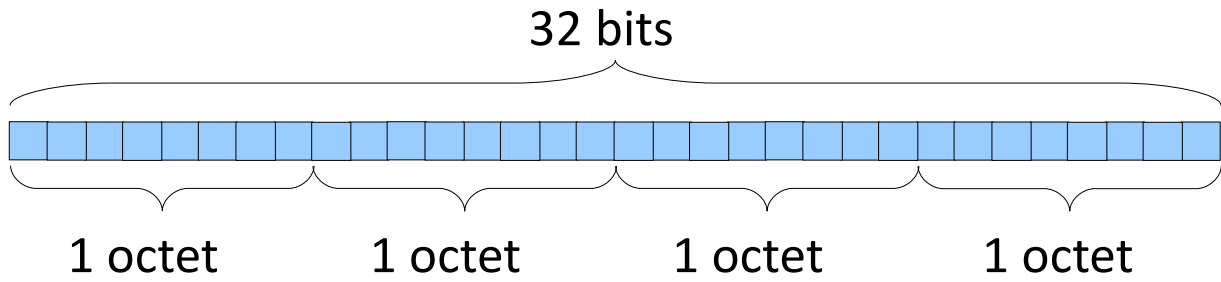
- compter sur ses doigts en binaire, jusqu'à  $2^{10}$
- faire des estimations de nombres données en binaire :  $2^{10} = 1024$  donc  $2^{10} \approx 1000$

$2^{32} \approx 4$  milliards

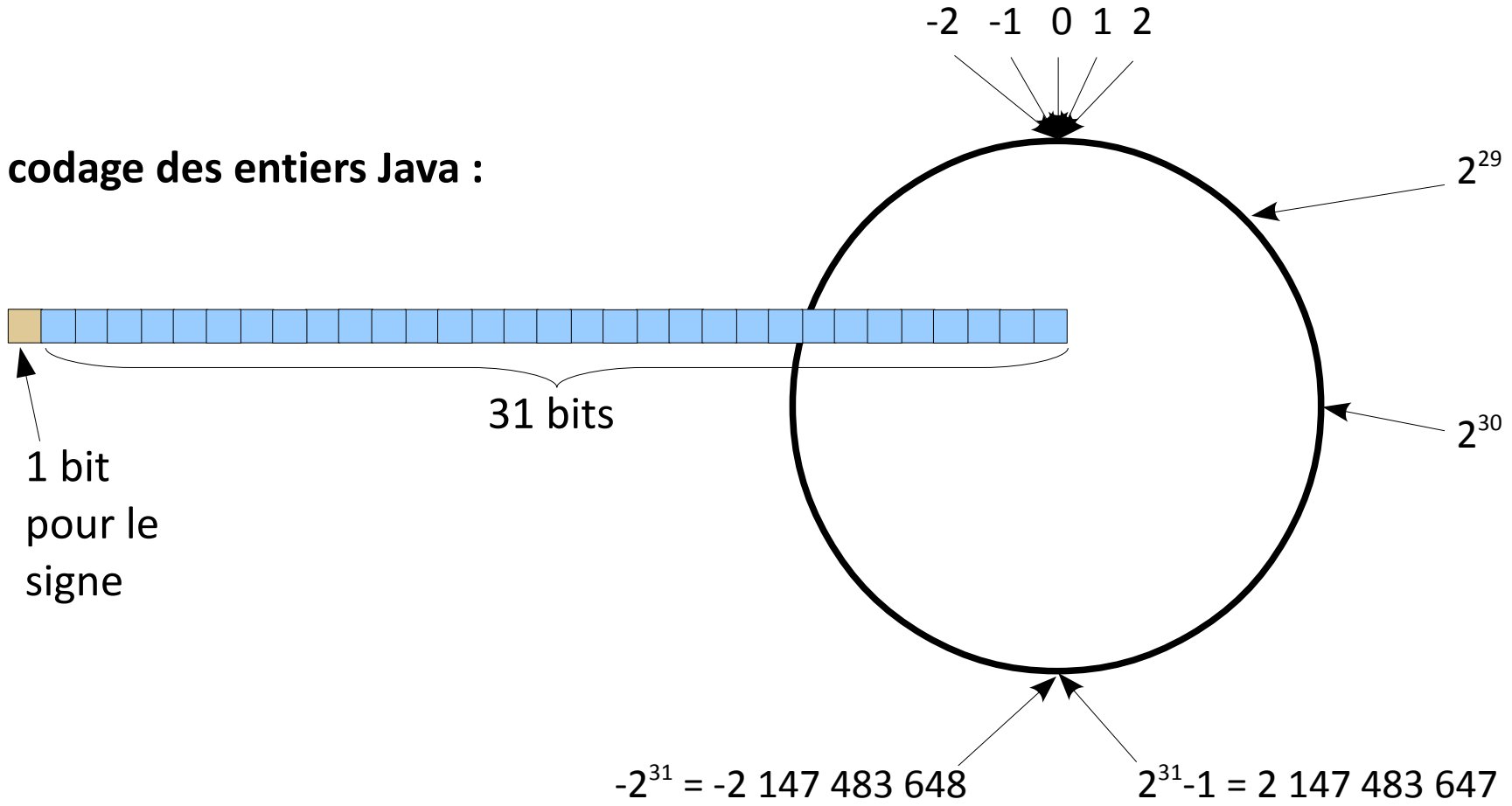
$2^{32} = 4\ 294\ 967\ 296$

# Codage des entiers 32 bits

- 1 bit = 0 ou 1
- 1 octet = 8 bits
- 1 Ko (kiloctet) = 1024 octets
- 1 Mo (mégaoctet) = 1024 Ko (disquette)
- 1 Go (gigaoctet) = 1024 Mo (carte mémoire, 2h de vidéo en DivX)
- 1 To (téraoctet) = 1024 Go (disque dur externe)



## Le codage des entiers Java :



# Autres codages

## • Chaînes de caractères

- ASCII : 7 bits, caractères simples codés de 32 à 127
- ANSI : 8 bits, caractères simples codés de 32 à 127, caractères accentués de 128 à 255
- UTF-8 : de 1 à 4 octets

## • Couleurs d'une image

- RGB : "red, blue, green", 1 octet pour chacun :
- valeurs entre 0 et 255
- codage hexadécimal avec 2 symboles

### *La "minute culturelle"*

Hexadécimal : en base 16 (ἕξάς : six, decem : dix)

Codé par les chiffres de 0 à 9 et les lettres A B C D E F

	A	B	C	D	E	F
	↑	↑	↑	↑	↑	↑
	10	11	12	13	14	15

- Deux symboles pour un octet :  
 $16^2$  valeurs possibles = 256

- Utilisé pour coder les couleurs en HTML :  
couleur="#RRGGBB"  
rouge="#FF0000", vert="#00FF00"  
#800080 ?

# Les booléens

- Opérations sur les booléens :  
et, ou, non

<b>ET</b>	VRAI	FAUX
VRAI	VRAI	FAUX
FAUX	FAUX	FAUX

<b>OU</b>	VRAI	FAUX
VRAI	VRAI	VRAI
FAUX	VRAI	FAUX

<b>NON</b>	VRAI	FAUX
	FAUX	VRAI

# Les opérations de base en Java

---

- **Type entier `int`**

`+`, `-`, `*`, `/` (division entière), `%` (reste modulo), `^` (puissance)

- **Type flottant `float`, ou `double` (plus précis)**

`+`, `-`, `*`, `/`

- **Type booléen `boolean`**

`false` (faux), `true` (vrai), `&&` (et), `||` (ou), `!` (non)

- **Type chaîne de caractères `String`**

`+` (concaténation : `"INF"+"120"="INF120"`)