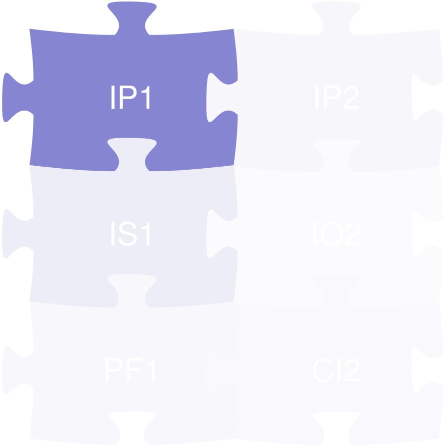
Principes de fonctionnement des machines binaires

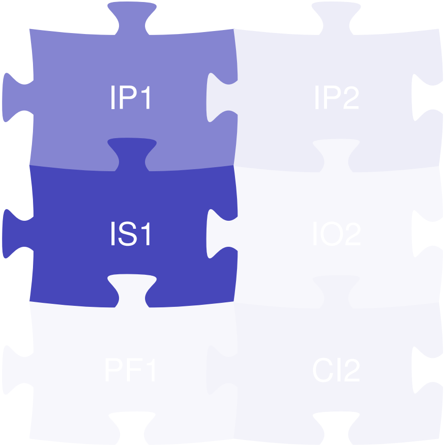
2020–2021

Matthieu Picantin

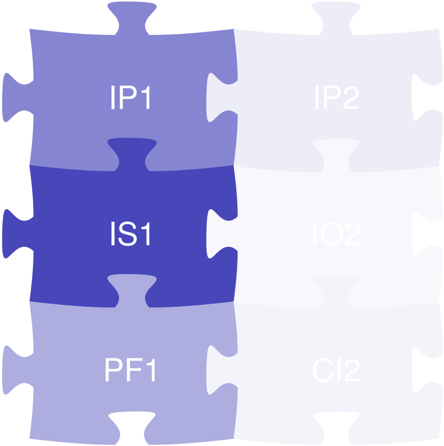
https://moodle.u-paris.fr/



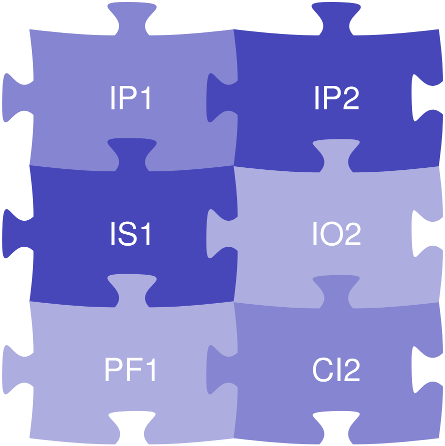
https://moodle.u-paris.fr/



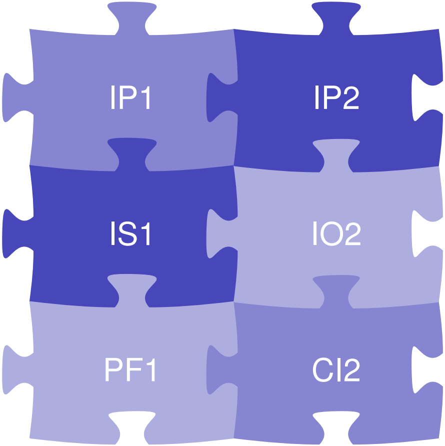
https://moodle.u-paris.fr/



https://moodle.u-paris.fr/



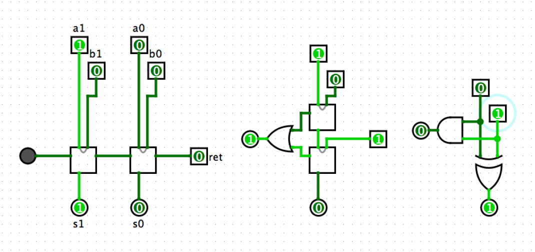
https://moodle.u-paris.fr/

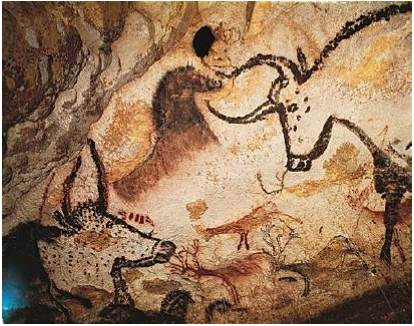


`https://moodle.u-paris.fr/`

- numération et arithmétique
- numération et arithmétique en machine
- numérisation et codage (texte, images)
- compression, cryptographie, contrôle d'erreur
- logique et calcul propositionnel
- circuits numériques

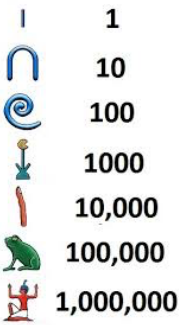
- numération et arithmétique
- numération et arithmétique en machine
- numérisation et codage (texte, images)
- compression, cryptographie, contrôle d'erreur
- logique et calcul propositionnel
- circuits numériques





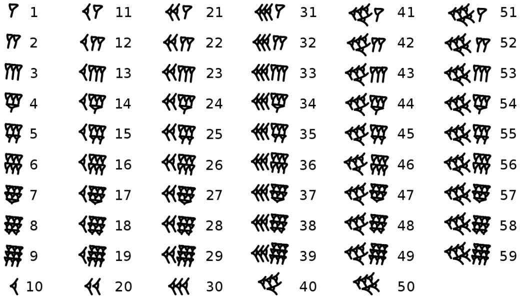
Numération égyptienne

- il y a 5000 ans environ
- numération additive
- sept hiéroglyphes



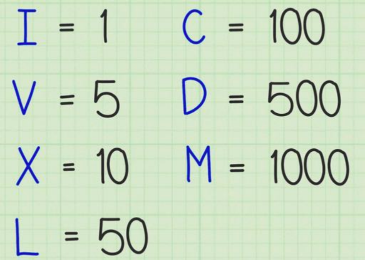
Numération babylonienne

- il y a 4000 ans environ
- numération positionnelle sexagésimale (base 60)
- 59 *chiffres* basés eux sur un système additif décimal



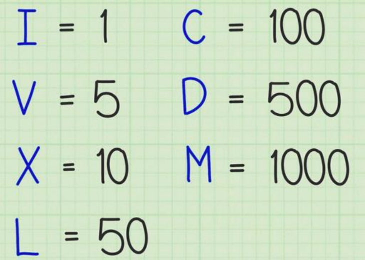
Numération romaine

- il y a 2800 ans environ
- additive et partiellement soustractive
- combinaison de sept *chiffres romains*
- utilisation moderne pour les ordinaux



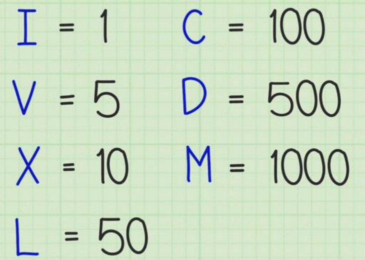
Numération romaine

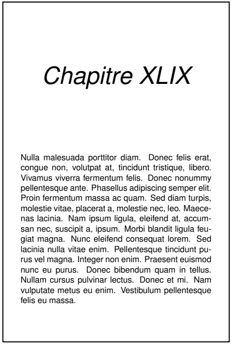
- il y a 2800 ans environ
- additive et partiellement soustractive
- combinaison de sept *chiffres romains*
- utilisation moderne pour les ordinaux



Numération romaine

- il y a 2800 ans environ
- additive et partiellement soustractive
- combinaison de sept *chiffres romains*
- utilisation moderne pour les ordinaux



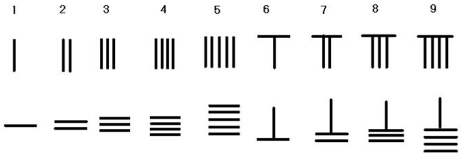


Chapitre XLIX

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

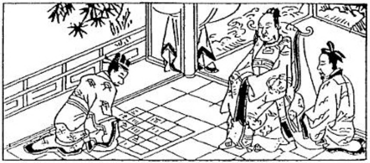
Numération chinoise à bâtons

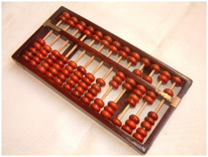
- il y a 2300 ans environ
- numération positionnelle décimale
- pas de chiffre zéro
- deux séries de neuf *chiffres bâtons*



(positions paires)

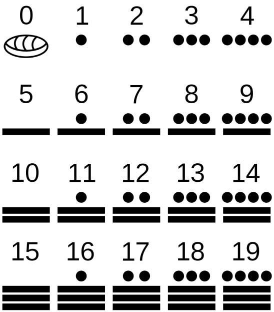
(positions impaires)





Numération maya

- il y a 2300 ans environ
- numération positionnelle vigécimale (base 20)
- 19 *chiffres* basés eux sur un système additif quinaire (base 5)
- chiffre zéro (deux *nombres zéros*: l'un cardinal, l'autre ordinal)



Numération positionnelle décimale

- dix chiffres, disons $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $(a_p \cdots a_0)_{10}$ représente le nombre $\displaystyle\sum_{k=0}^{p} a_k 10^k$, soit

$$a_p \times 10^p + a_{p-1} \times 10^{p-1} + \cdots + a_2 \times 10^2 + a_1 \times 10 + a_0$$

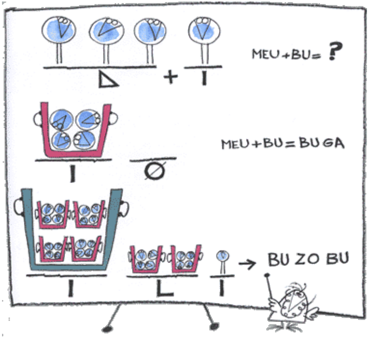
Hindu-Arabic	1	2	3	4	5	6	7	8	9	0
Arabic	١	٢	٣	٤	٥	٦	٧	٨	٩	٠
Devanagari (Hindi)	१	२	३	४	५	६	७	८	९	०
Tibetan	༡	༢	༣	༤	༥	༦	༧	༨	༩	༠
Bengali	১	২	৩	৪	৫	৬	৭	৮	৯	০
Thai	๑	๒	๓	๔	๕	๖	๗	๘	๙	๐

Numération positionnelle en base $b > 0$

- exactement b chiffres, disons $\{0, 1, \cdots, b-1\}$
- $(a_p \cdots a_0)_b$ représente le nombre $\sum_{k=0}^{p} a_k b^k$

Numération positionnelle en base $b > 0$

- exactement b chiffres, disons $\{0, 1, \cdots, b-1\}$
- $(a_p \cdots a_0)_b$ représente le nombre $\sum_{k=0}^{p} a_k b^k$



Comment convertir une écriture en base b $(a_p \cdots a_0)_b$

vers une écriture en base d? $(c_q \cdots c_0)_d$

Méthode par encadrements successifs (plutôt naïve)

- on encadre le nombre n entre deux facteurs successifs
  $$c_k d^k \leq n < (c_k + 1)d^k \qquad \text{(avec } 0 < c_k < d\text{)}$$
- on collecte le chiffre c_k (pour la position k)
- on recommence avec le nombre n auquel on a retranché $c_k d^k$
- on s'arrête quand le nombre est nul
  et on renvoie l'écriture avec les chiffres collectés

Comment convertir une écriture en base b $\qquad (a_p \cdots a_0)_b$

vers une écriture en base d? $\qquad (c_q \cdots c_0)_d$

Méthode par encadrements successifs (plutôt naïve)

- on encadre le nombre n entre deux facteurs successifs
  $c_k d^k \leq n < (c_k + 1) d^k$ (avec $0 < c_k < d$)
- on collecte le chiffre c_k (pour la position k)
- on recommence avec le nombre n auquel on a retranché $c_k d^k$
- on s'arrête quand le nombre est nul
  et on renvoie l'écriture avec les chiffres collectés

Comment convertir une écriture en base b $\quad (a_p \cdots a_0)_b$

vers une écriture en base d? $\quad (c_q \cdots c_0)_d$

Méthode par divisions successives (vrai algo)

- on divise n par la base d (calcul fait en base b)
- on collecte le reste r
- on recommence avec le quotient q
- on s'arrête quand le nombre est nul

  et on renvoie la suite **inversée** des restes collectés

$$\begin{array}{c|c} n & d \\ \hline r & q \end{array}$$

Comment convertir une écriture en base b $(a_p \cdots a_0)_b$

vers une écriture en base d? $(c_q \cdots c_0)_d$

Méthode par divisions successives (vrai algo)

à privilégier quand la base b est confortable

- on divise n par la base d (calcul fait en base b)
- on collecte le reste r
- on recommence avec le quotient q
- on s'arrête quand le nombre est nul

  et on renvoie la suite **inversée** des restes collectés

$$\begin{array}{c|c} n & d \\ \cline{2-2} r & q \end{array}$$

Comment convertir une écriture en base b $\qquad (a_p \cdots a_0)_b$

vers une écriture en base d? $\qquad (c_q \cdots c_0)_d$

Méthode par recomposition – méthode *de Horner*

- on utilise la définition $\displaystyle\sum_{k=0}^{p} a_k b^k$ en calculant dans la base d
- on peut diminuer le nombre d'opérations en utilisant

$$
\begin{array}{rl}
 & a_p \times b^p + a_{p-1} \times b^{p-1} + \cdots + a_2 \times b^2 + a_1 \times b + a_0 \\
= & (a_p \times b^{p-1} + a_{p-1} \times b^{p-2} + \cdots + a_2 \times b + a_1) \times b + a_0 \\
\vdots & \\
= & (\cdots(a_p) \times b + a_{p-1}) \times b + \cdots + a_2) \times b + a_1) \times b + a_0
\end{array}
$$

Comment convertir une écriture en base b $(a_p \cdots a_0)_b$

vers une écriture en base d? $(c_q \cdots c_0)_d$

Méthode par recomposition – méthode *de Horner*

à privilégier quand la base d est confortable

- on utilise la définition $\sum_{k=0}^{p} a_k b^k$ en calculant dans la base d
- on peut diminuer le nombre d'opérations en utilisant

$$
\begin{aligned}
& a_p \times b^p + a_{p-1} \times b^{p-1} + \cdots + a_2 \times b^2 + a_1 \times b + a_0 \\
= \ & (a_p \times b^{p-1} + a_{p-1} \times b^{p-2} + \cdots + a_2 \times b + a_1) \times b + a_0 \\
\vdots \ & \\
= \ & (\cdots(a_p) \times b + a_{p-1}) \times b + \cdots + a_2) \times b + a_1) \times b + a_0
\end{aligned}
$$

Comment convertir une écriture en base b^s $(a_p \cdots a_0)_{b^s}$

vers une écriture en base b^t? $(c_q \cdots c_0)_{b^t}$

Méthode par dépliage-repliage

- on écrit chaque chiffre de l'écriture en base b^s dans la base b

  $a_i = a_{i_1} \cdots a_{i_s}$
- on regroupe par paquet de t chiffres **en commençant par la droite**
- on convertit chaque paquet dans la base b^t

$$\begin{array}{ccc} & ((a_{p,1} \cdots a_{p,s}) \cdots (a_{0,1} \cdots a_{0,s}))_b & \\ \text{dépliage} \nearrow & & \searrow \text{repliage} \\ (a_p \cdots a_0)_{b^s} & & (c_q \cdots c_0)_{b^t} \end{array}$$

Comment convertir une écriture en base b^s $\quad (a_p \cdots a_0)_{b^s}$

vers une écriture en base b^t ? $\quad (c_q \cdots c_0)_{b^t}$

Méthode par dépliage-repliage

- on écrit chaque chiffre de l'écriture en base b^s dans la base b

  $$a_i = a_{i_1} \cdots a_{i_s}$$

- on regroupe par paquet de t chiffres **en commençant par la droite**
- on convertit chaque paquet dans la base b^t

$$((a_{p,1} \cdots a_{p,s}) \cdots (a_{0,1} \cdots a_{0,s}))_b$$

dépliage $\quad$ repliage

$$(a_p \cdots a_0)_{b^s} \qquad (c_q \cdots c_0)_{b^t}$$

Comment convertir une écriture en base b^s $(a_p \cdots a_0)_{b^s}$

vers une écriture en base b^t? $(c_q \cdots c_0)_{b^t}$

Méthode par dépliage-repliage

- on écrit chaque chiffre de l'écriture en base b^s dans la base b

  $a_i = a_{i_1} \cdots a_{i_s}$
- on regroupe par paquet de t chiffres **en commençant par la droite**
- on convertit chaque paquet dans la base b^t

$$((a_{p,1} \cdots a_{p,s}) \cdots (a_{0,1} \cdots a_{0,s}))_b$$

dépliage

repliage

$$(a_p \cdots a_0)_{b^s} \qquad (c_q \cdots c_0)_{b^t}$$

Comment convertir une écriture en base b^s $\quad (a_p \cdots a_0)_{b^s}$

vers une écriture en base b^t? $\quad (c_q \cdots c_0)_{b^t}$

Méthode par dépliage-repliage

- on écrit chaque chiffre de l'écriture en base b^s dans la base b

  $$a_i = a_{i_1} \cdots a_{i_s}$$

- on regroupe par paquet de t chiffres **en commençant par la droite**
- on convertit chaque paquet dans la base b^t

$$((a_{p,1} \cdots a_{p,s}) \cdots (a_{0,1} \cdots a_{0,s}))_b$$

dépliage $\nearrow$ $\quad\quad$ $\searrow$ repliage

$$(a_p \cdots a_0)_{b^s} \qquad\qquad (c_q \cdots c_0)_{b^t}$$

Comment convertir une écriture en base b^s $\quad (a_p \cdots a_0)_{b^s}$

vers une écriture en base b^t ? $\quad (c_q \cdots c_0)_{b^t}$

Méthode par dépliage-repliage

- on écrit chaque chiffre de l'écriture en base b^s dans la base b

  $a_i = a_{i_1} \cdots a_{i_s}$
- on regroupe par paquet de t chiffres **en commençant par la droite**
- on convertit chaque paquet dans la base b^t

$$\begin{array}{ccc} & ((a_{p,1} \cdots a_{p,s}) \cdots (a_{0,1} \cdots a_{0,s}))_b & \\ \text{dépliage} \nearrow & & \searrow \text{repliage} \\ (a_p \cdots a_0)_{b^s} & & (c_q \cdots c_0)_{b^t} \end{array}$$

Comment convertir une écriture en base b^s $\quad (a_p \cdots a_0)_{b^s}$

vers une écriture en base b^t ? $\quad (c_q \cdots c_0)_{b^t}$

Méthode par dépliage-repliage

à privilégier face aux autres méthodes

- on écrit chaque chiffre de l'écriture en base b^s dans la base b

  $a_i = a_{i_1} \cdots a_{i_s}$
- on regroupe par paquet de t chiffres **en commençant par la droite**
- on convertit chaque paquet dans la base b^t

$$
\begin{tikzcd}
 & ((a_{p,1} \cdots a_{p,s}) \cdots (a_{0,1} \cdots a_{0,s}))_b \arrow[dr, "\text{repliage}"] & \\
(a_p \cdots a_0)_{b^s} \arrow[ur, "\text{dépliage}"] & & (c_q \cdots c_0)_{b^t}
\end{tikzcd}
$$

Comment convertir une écriture en base b $\quad (a_p \cdots a_0)_b$

vers une écriture en base d? $\quad (c_q \cdots c_0)_d$

Méthode par transducteur

- on construit un transducteur avec
  - d états numérotés $\{0, 1, \ldots, d-1\}$
  - pour chaque état s et chaque lettre x dans $\{0, 1, \ldots, b-1\}$

$$\begin{tikzcd} s \arrow[r, "x \mid y"] & r \end{tikzcd}$$

    avec r le reste et y le quotient de la division de $s \times b + x$ par d

- pour chaque écriture en base b à convertir en base d
  - on fait lire le mot depuis l'état de départ 0
  - on collecte l'état d'arrivée
  - on recommence avec le mot obtenu en sortie
  - on s'arrête quand le mot devient vide
    et on renvoie la suite inversée des états d'arrivée collectés

Comment convertir une écriture en base b $\qquad (a_p \cdots a_0)_b$

vers une écriture en base d? $\qquad (c_q \cdots c_0)_d$

Méthode par transducteur

hors programme!
(rendez-vous en S3)

- on construit un transducteur avec
  - d états numérotés $\{0, 1, \ldots, d-1\}$
  - pour chaque état s et chaque lettre x dans $\{0, 1, \ldots, b-1\}$

    $$s \xrightarrow{x \mid y} r$$

    avec r le reste et y le quotient de la division de $s \times b + x$ par d
- pour chaque écriture en base b à convertir en base d
  - on fait lire le mot depuis l'état de départ 0
  - on collecte l'état d'arrivée
  - on recommence avec le mot obtenu en sortie
  - on s'arrête quand le mot devient vide
    et on renvoie la suite **inversée** des états d'arrivée collectés

Comment convertir une écriture en base b $(a_p \cdots a_0)_b$

vers une écriture en base d? $(c_q \cdots c_0)_d$

Méthode par transducteur

hors programme!
(rendez-vous en S3)

- on construit un transducteur avec
  - d états numérotés $\{0, 1, \ldots, d-1\}$
  - pour chaque état s et chaque lettre x dans $\{0, 1, \ldots, b-1\}$

$$s \xrightarrow{x \mid y} r$$

avec r le reste et y le quotient de la division de $s \times b + x$ par d

- pour chaque écriture en base b à convertir en base d
  - on fait lire le mot depuis l'état de départ 0
  - on collecte l'état d'arrivée
  - on recommence avec le mot obtenu en sortie
  - on s'arrête quand le mot devient vide
    et on renvoie la suite **inversée** des états d'arrivée collectés

Comment convertir une écriture en base b $\qquad (a_p \cdots a_0)_b$

vers une écriture en base d? $\qquad (c_q \cdots c_0)_d$

Méthode par transducteur

hors programme!
(rendez-vous en S3)

- on construit un transducteur avec
  - d états numérotés $\{0, 1, \ldots, d-1\}$
  - pour chaque état s et chaque lettre x dans $\{0, 1, \ldots, b-1\}$

    $$s \xrightarrow{x \mid y} r$$

    avec r le reste et y le quotient de la division de $s \times b + x$ par d
- pour chaque écriture en base b à convertir en base d
  - on fait lire le mot depuis l'état de départ 0
  - on collecte l'état d'arrivée
  - on recommence avec le mot obtenu en sortie
  - on s'arrête quand le mot devient vide
    et on renvoie la suite **inversée** des états d'arrivée collectés