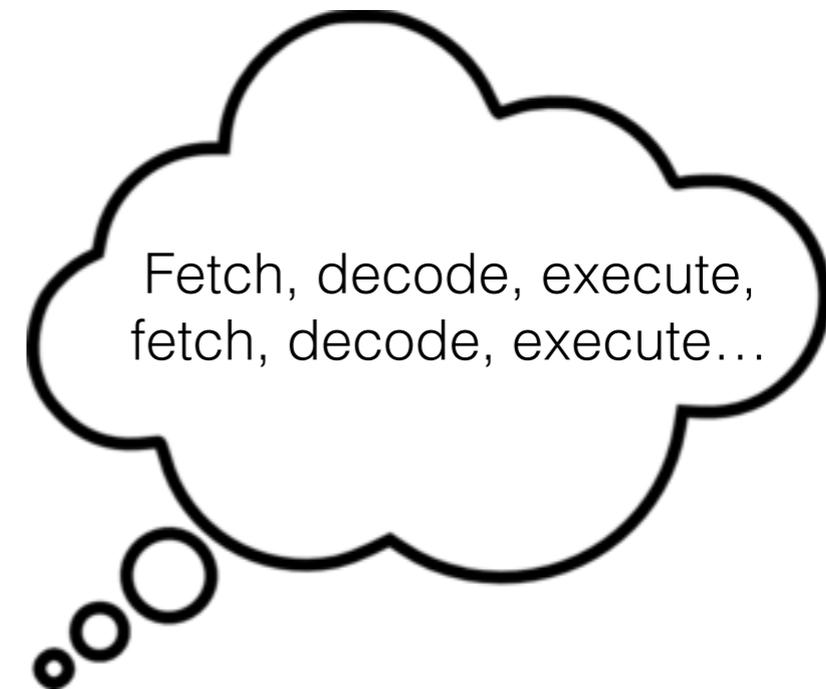
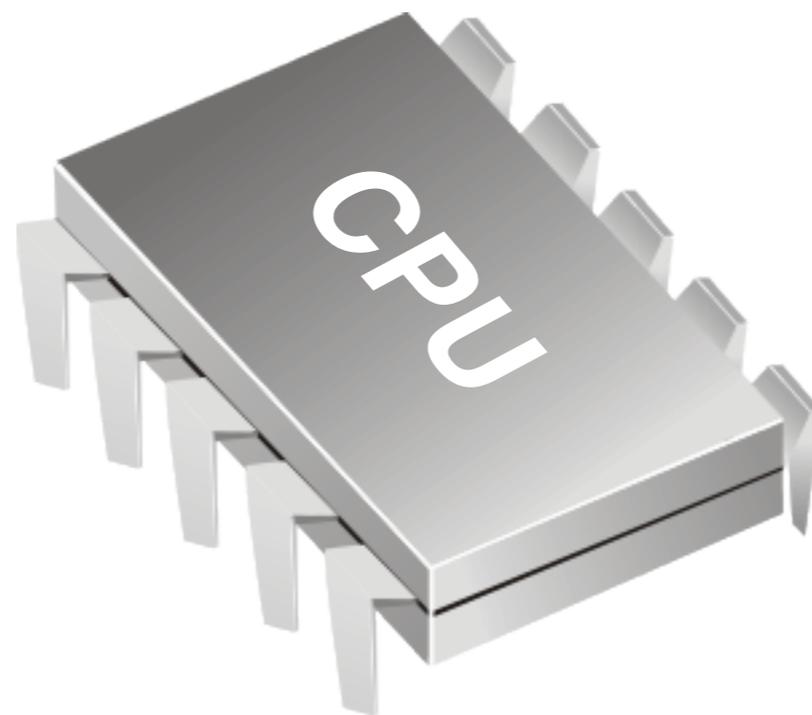


# Introduction aux microprocesseurs



# Qu'est-ce qu'un microprocesseur?

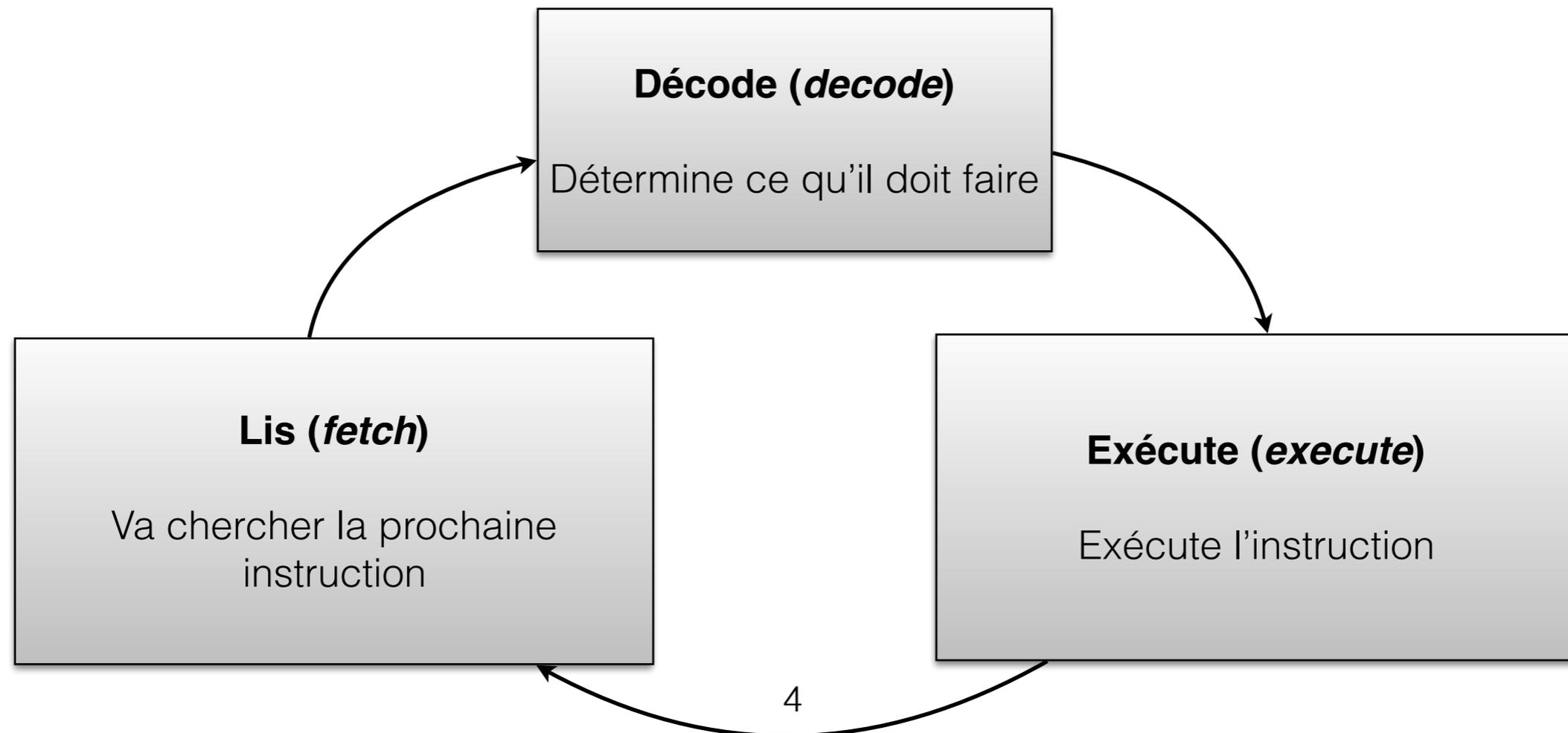
C'est l'unité centrale d'un ordinateur, qui exécute les instructions et traite les données des programmes.

Durant le cours, nous utiliserons également le terme CPU (*Central Processing Unit*) pour désigner le microprocesseur.

Qu'est-ce que fait un  
microprocesseur?

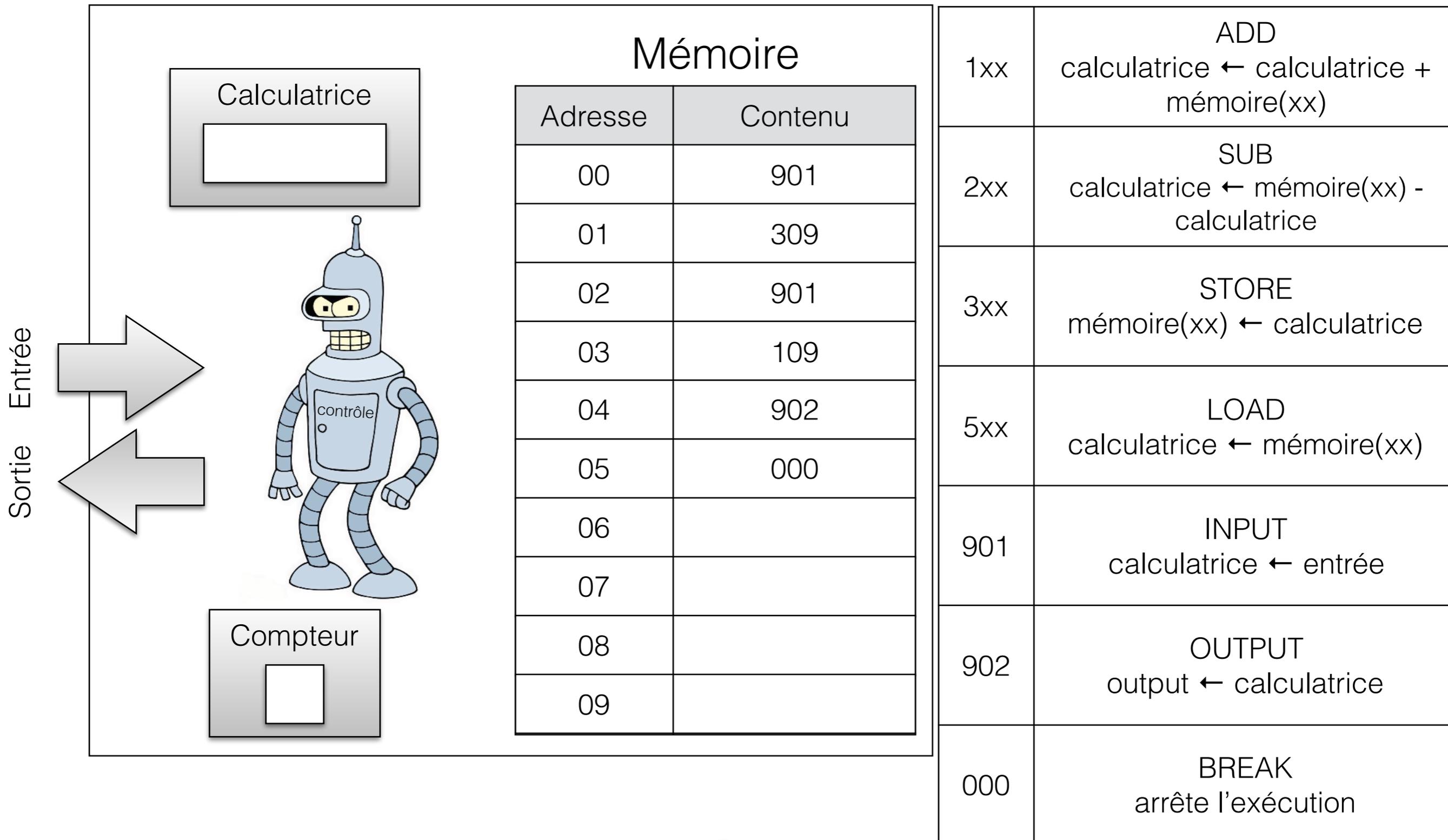
# Cycle d'instructions

- Que fait le microprocesseur?
- Seulement trois choses:
  - Lis (*fetch*): va chercher la prochaine instruction
  - Décode (*decode*): détermine ce qu'il doit faire, à partir de l'instruction
  - Exécute (*execute*): exécute l'instruction



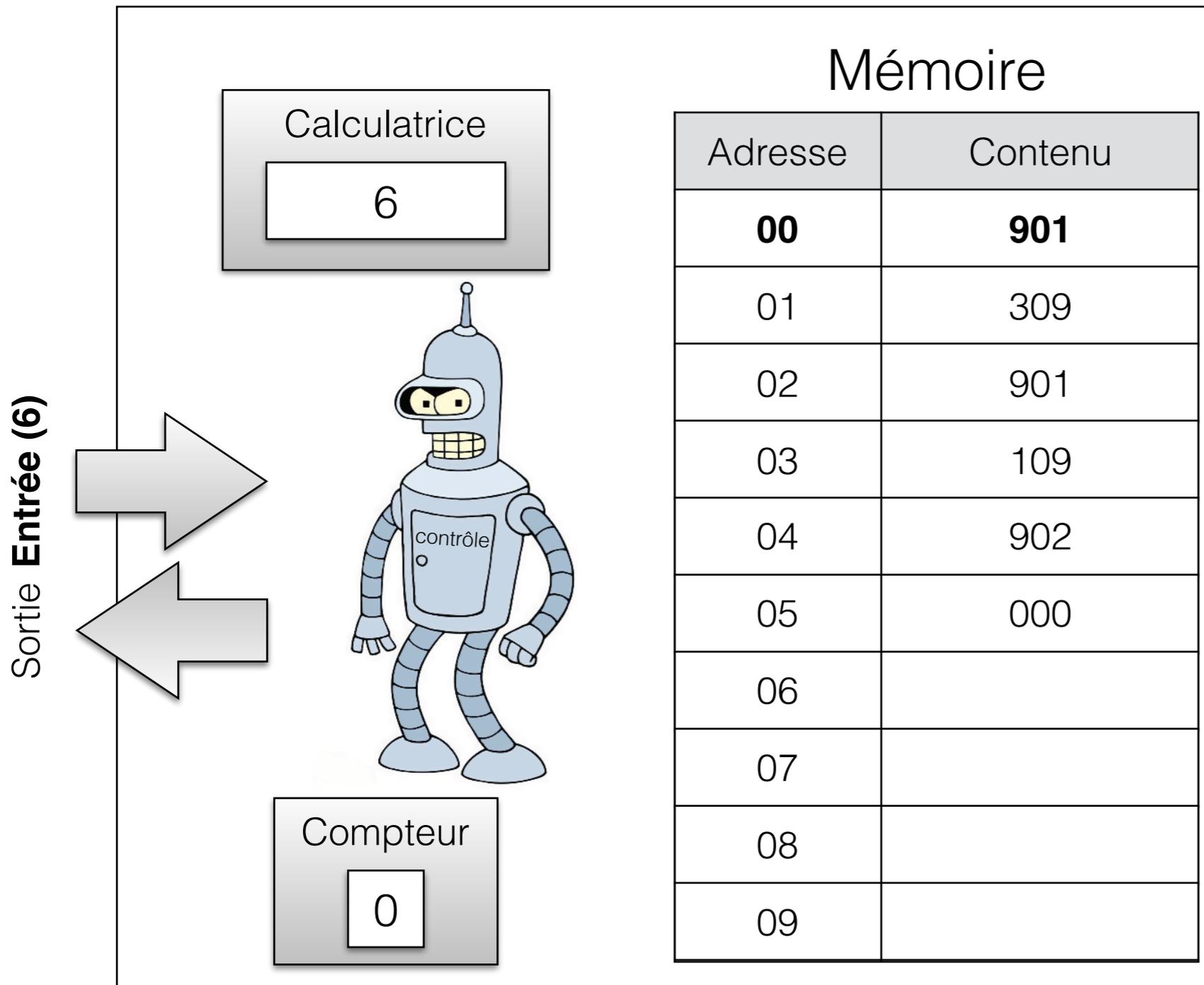
# Qu'est-ce que fait un microprocesseur?

Instructions (3 chiffres)



Résultats de l'exercice  
précédent, étape par étape

# Qu'est-ce que fait un microprocesseur?



## Décodage

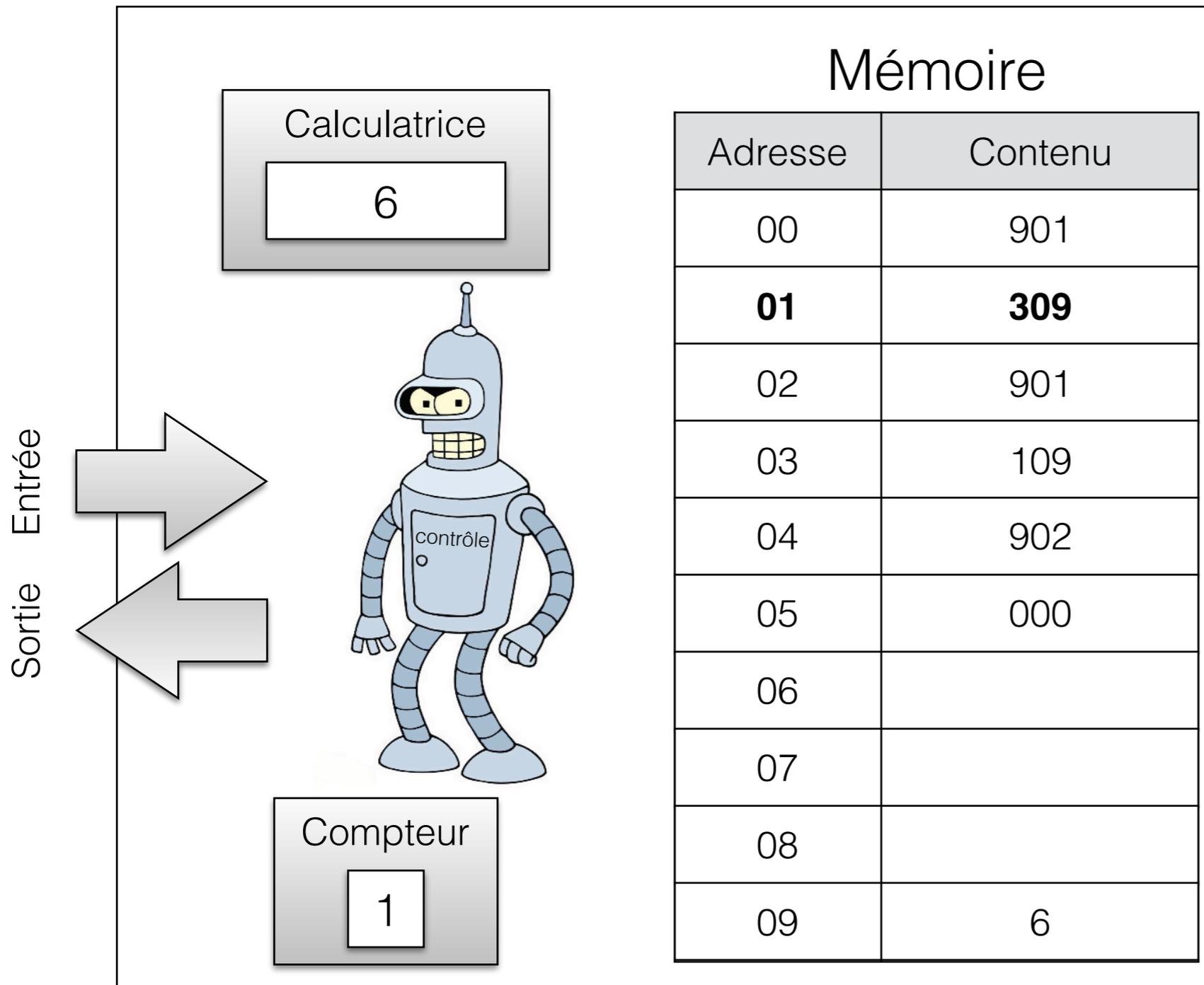
L'instruction est 901, ce qui correspond à INPUT.

L'unité de contrôle prend ce qui est donné par l'utilisateur en entrée et le place dans la calculatrice.

Par exemple, supposons que l'utilisateur ait écrit la valeur "6" dans l'entrée.

1xx	ADD calculatrice ← calculatrice + mémoire(xx)
2xx	SUB calculatrice ← mémoire(xx) - calculatrice
3xx	STORE mémoire(xx) ← calculatrice
5xx	LOAD calculatrice ← mémoire(xx)
<b>901</b>	<b>INPUT</b> <b>calculatrice ← entrée</b>
902	OUTPUT output ← calculatrice
000	BREAK arrête l'exécution

# Qu'est-ce que fait un microprocesseur?



## Décodage

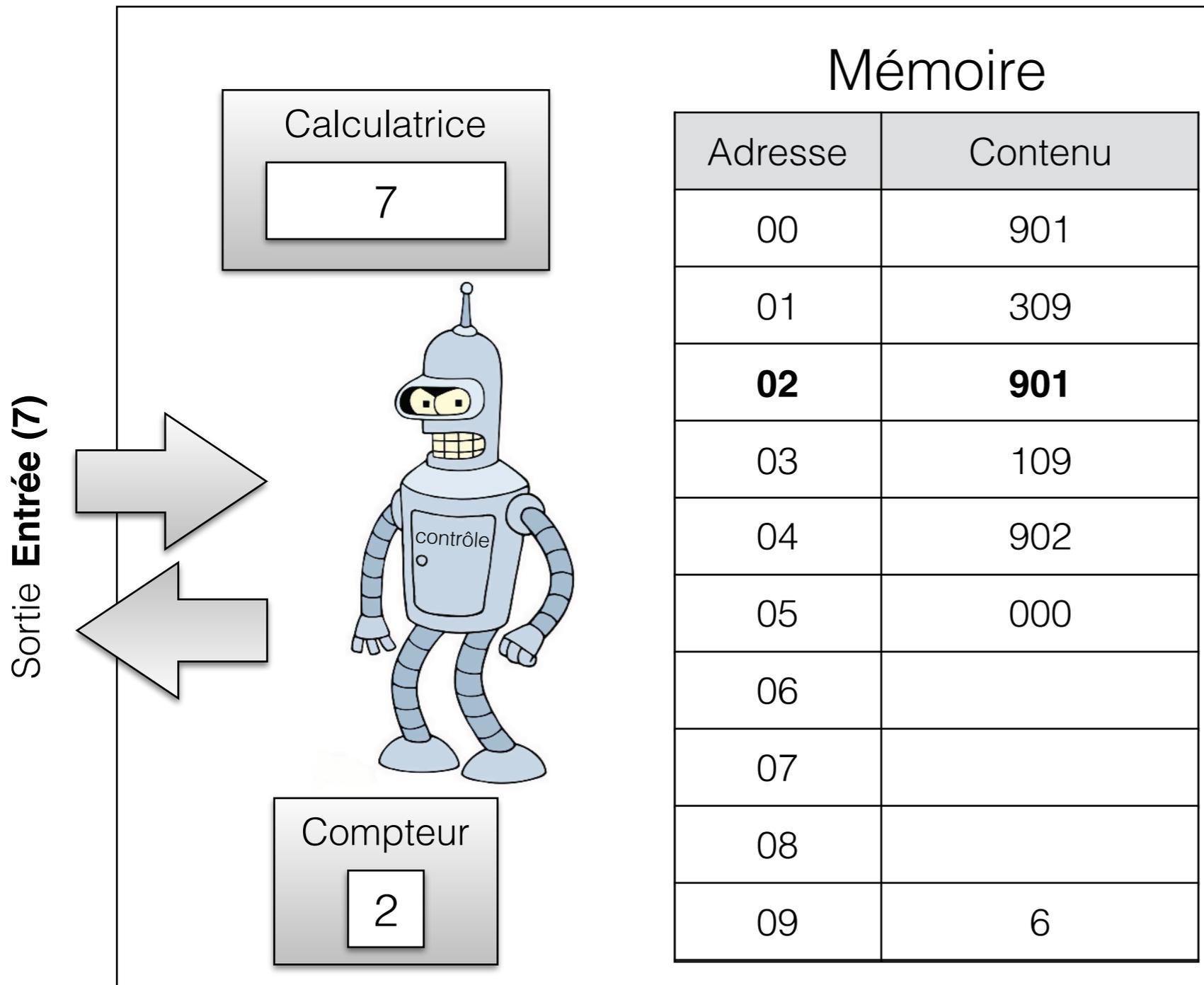
L'instruction est 309. Le premier chiffre (3) indique que l'instruction est un STORE.

Les deux chiffres suivants indiquent que le « xx » est 09.

L'unité de contrôle prend ce qui est dans la calculatrice, et le place en mémoire à l'adresse 09.

1xx	ADD calculatrice ← calculatrice + mémoire(xx)
2xx	SUB calculatrice ← mémoire(xx) - calculatrice
<b>3xx</b>	<b>STORE</b> <b>mémoire(xx) ← calculatrice</b>
5xx	LOAD calculatrice ← mémoire(xx)
901	INPUT calculatrice ← entrée
902	OUTPUT output ← calculatrice
000	BREAK arrête l'exécution

# Qu'est-ce que fait un microprocesseur?



## Décodage

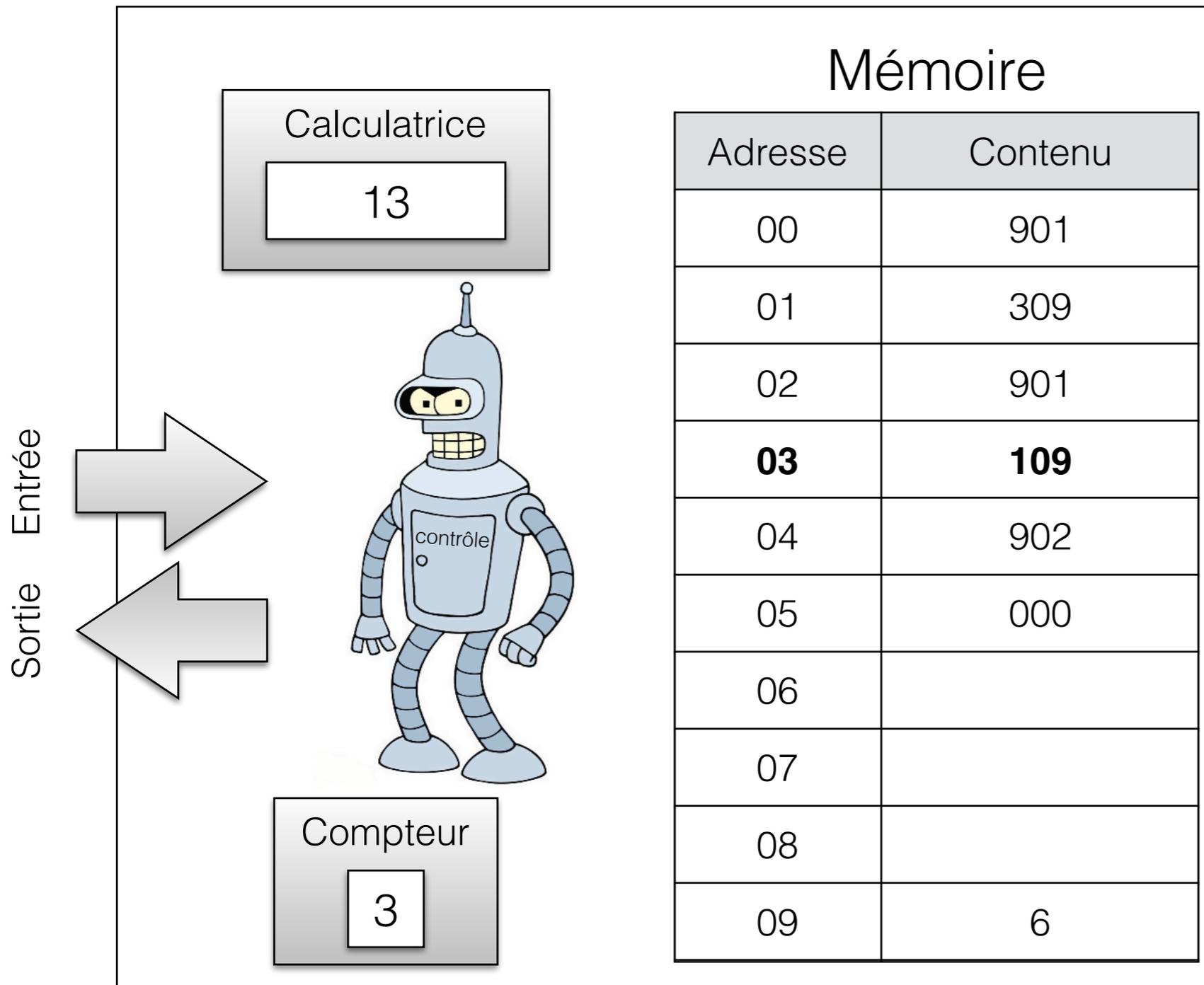
L'instruction est 901, ce qui correspond à INPUT.

L'unité de contrôle prend ce qui est donné par l'utilisateur en entrée et le place dans la calculatrice.

Par exemple, supposons que l'utilisateur ait écrit la valeur "7" dans l'entrée.

1xx	ADD calculatrice ← calculatrice + mémoire(xx)
2xx	SUB calculatrice ← mémoire(xx) - calculatrice
3xx	STORE mémoire(xx) ← calculatrice
5xx	LOAD calculatrice ← mémoire(xx)
<b>901</b>	<b>INPUT</b> <b>calculatrice ← entrée</b>
902	OUTPUT output ← calculatrice
000	BREAK arrête l'exécution

# Qu'est-ce que fait un microprocesseur?



## Décodage

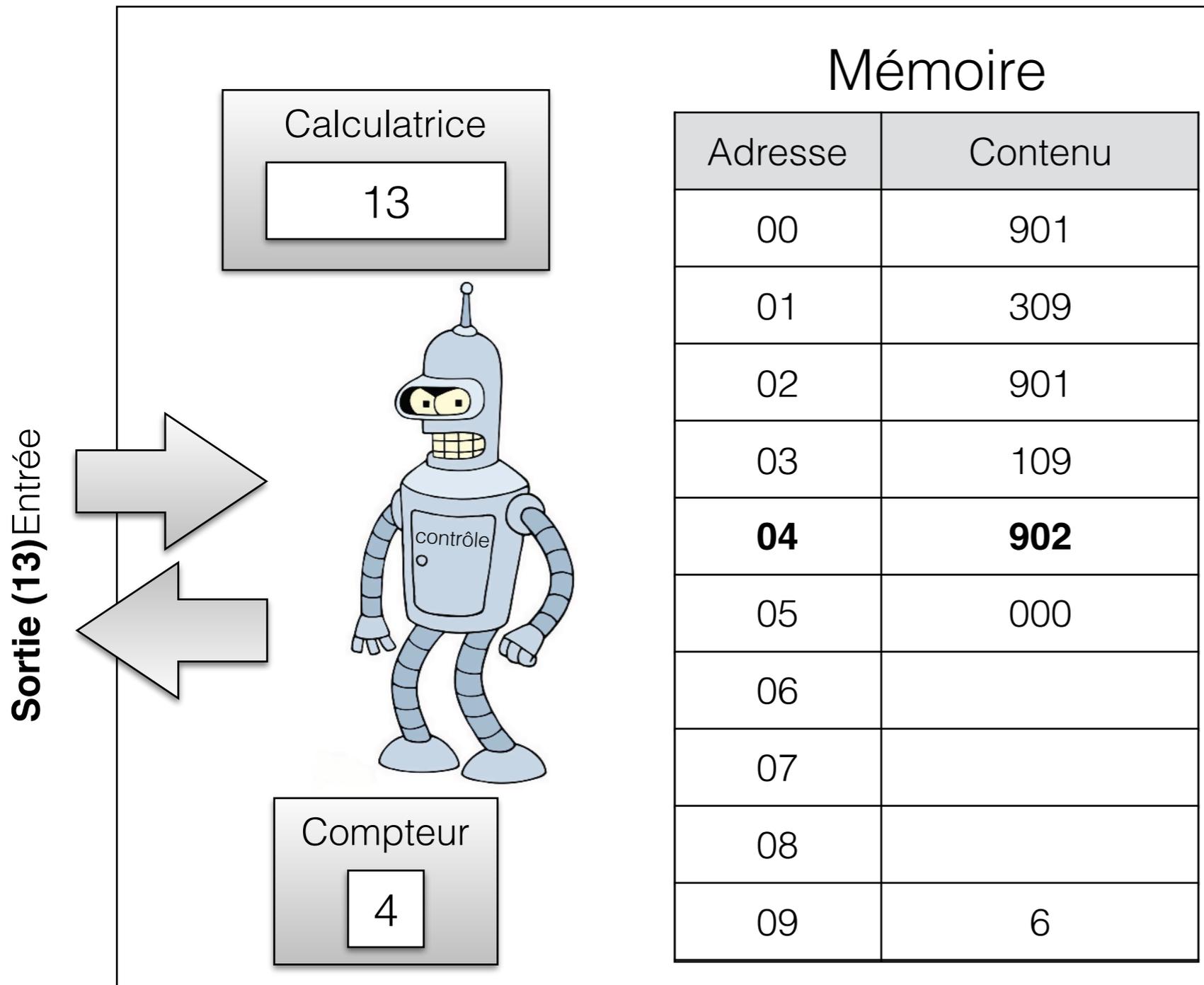
L'instruction est 109. Le premier chiffre (1) indique que l'instruction est un ADD.

Les deux chiffres suivants indiquent que le « xx » est 09.

L'unité de contrôle prend ce qui est dans la calculatrice, l'additionne à ce qui est en mémoire à l'adresse 09, et stocke le résultat dans la calculatrice.

<b>1xx</b>	<b>ADD</b> <b>calculatrice ← calculatrice +</b>
2xx	SUB calculatrice ← mémoire(xx) - calculatrice
3xx	STORE mémoire(xx) ← calculatrice
5xx	LOAD calculatrice ← mémoire(xx)
901	INPUT calculatrice ← entrée
902	OUTPUT output ← calculatrice
000	BREAK arrête l'exécution

# Qu'est-ce que fait un microprocesseur?



## Décodage

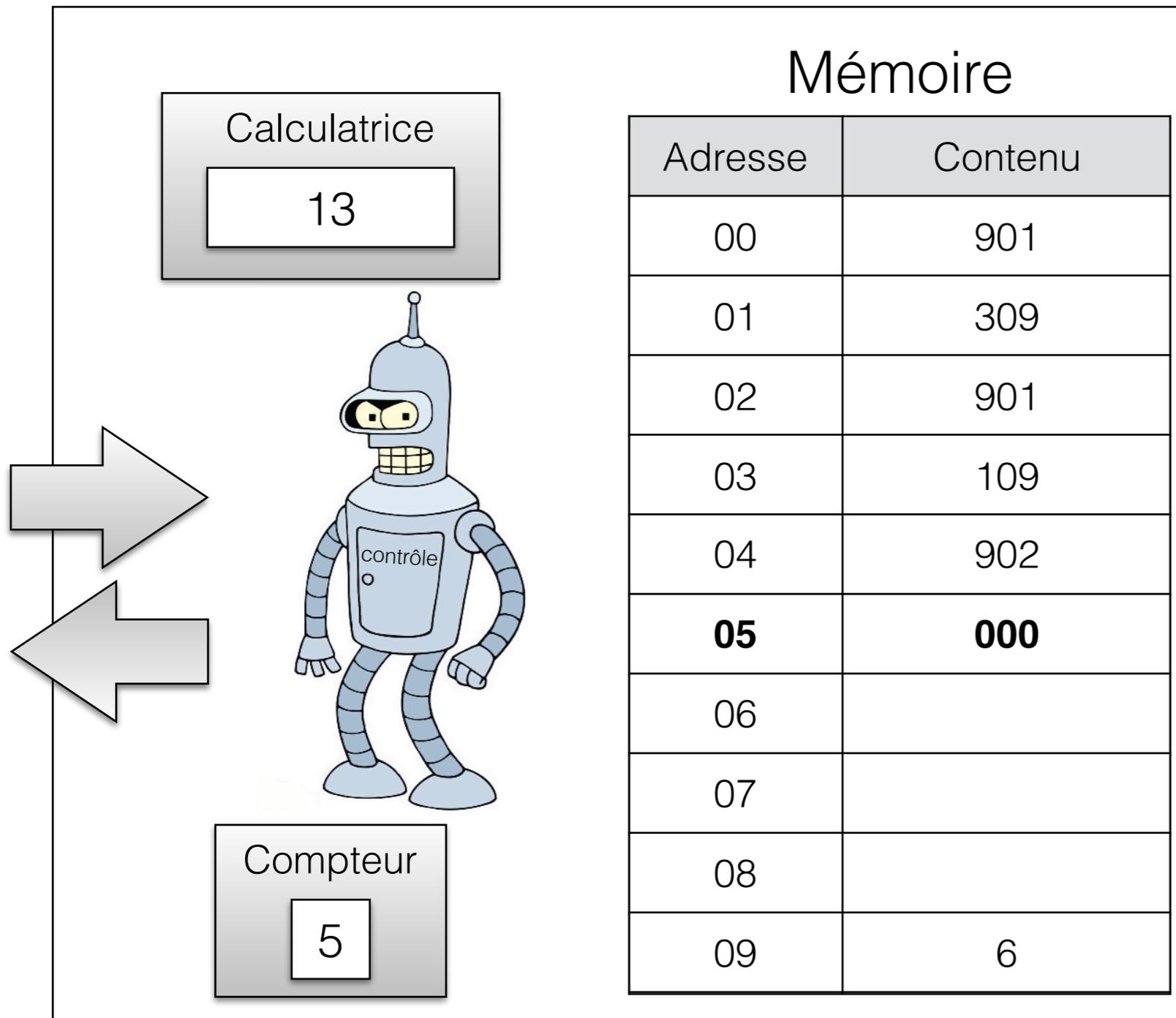
L'instruction est 902, ce qui correspond à OUTPUT.

L'unité de contrôle prend ce qui est dans la calculatrice et le donne à l'utilisateur via la sortie.

1xx	ADD calculatrice ← calculatrice + mémoire(xx)
2xx	SUB calculatrice ← mémoire(xx) - calculatrice
3xx	STORE mémoire(xx) ← calculatrice
5xx	LOAD calculatrice ← mémoire(xx)
901	INPUT calculatrice ← entrée
<b>902</b>	<b>OUTPUT</b> <b>output ← calculatrice</b>
000	BREAK arrête l'exécution

# Qu'est-ce que fait un microprocesseur?

Sortie (13) Entrée



## Décodage

L'instruction est 000, ce qui correspond à BREAK.

L'unité de contrôle arrête l'exécution de l'ordinateur.

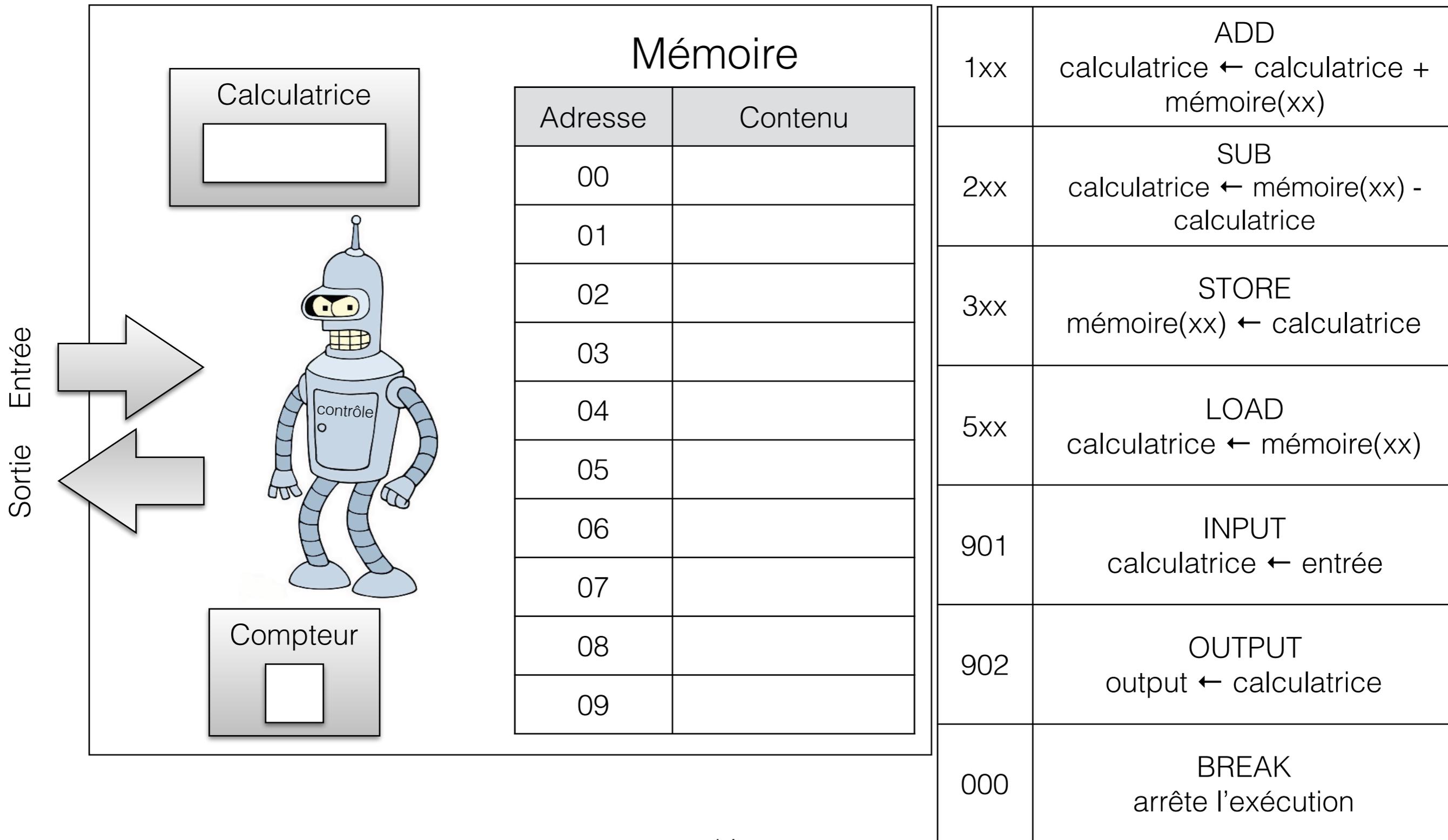
Cet ordinateur ne ferait alors plus rien, et il faudrait le redémarrer pour recommencer le programme.

1xx	ADD calculatrice ← calculatrice + mémoire(xx)
2xx	SUB calculatrice ← mémoire(xx) - calculatrice
3xx	STORE mémoire(xx) ← calculatrice
5xx	LOAD calculatrice ← mémoire(xx)
901	INPUT calculatrice ← entrée
902	OUTPUT output ← calculatrice
<b>000</b>	<b>BREAK</b> <b>arrête l'exécution</b>

**Donc: le programme précédent  
prend deux entrées de l'utilisateur,  
calcule leur somme,  
et retourne le résultat en sortie.**

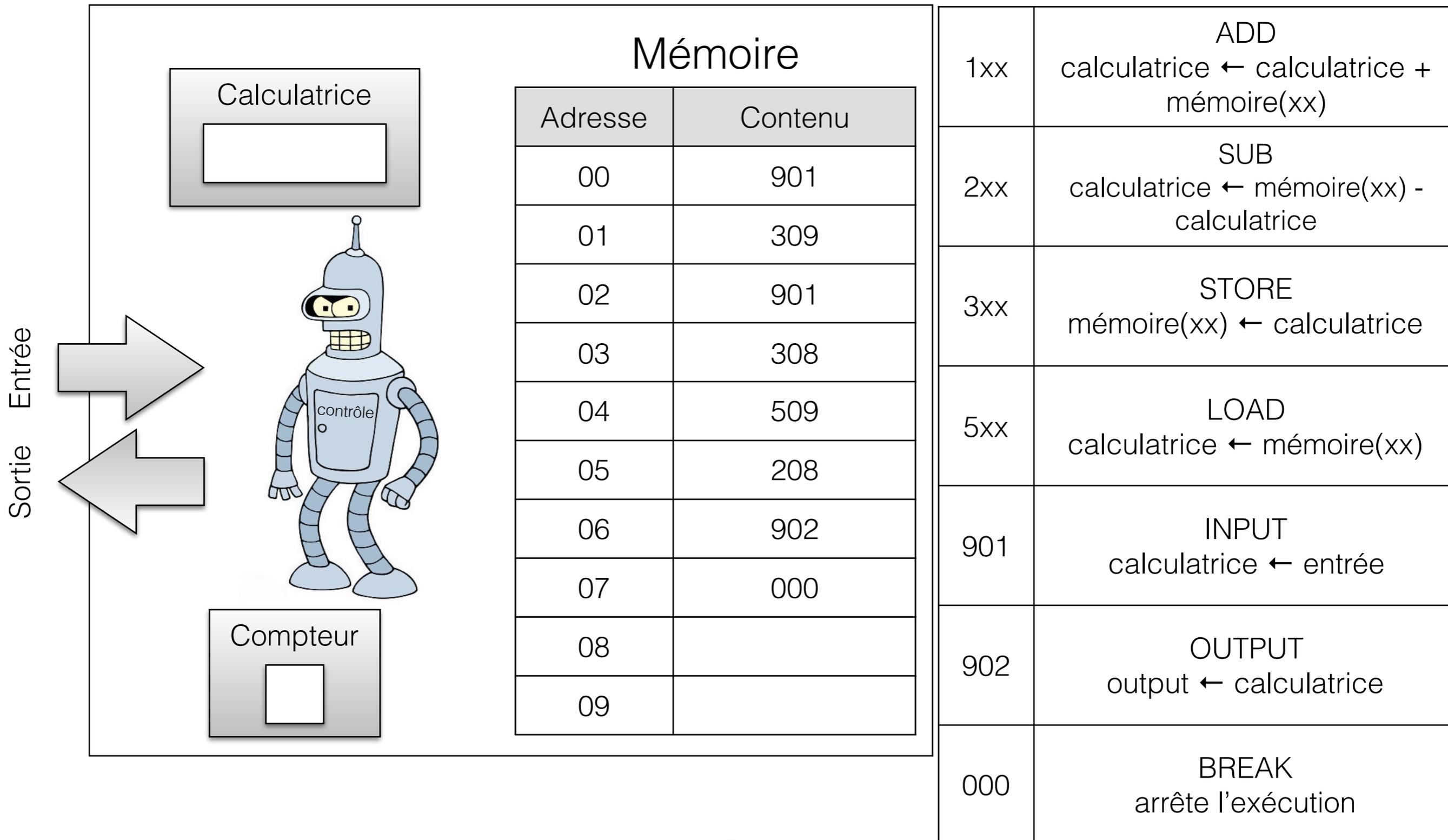
# Programme: sortie = entrée2 - entrée1?

Instructions (3 chiffres)



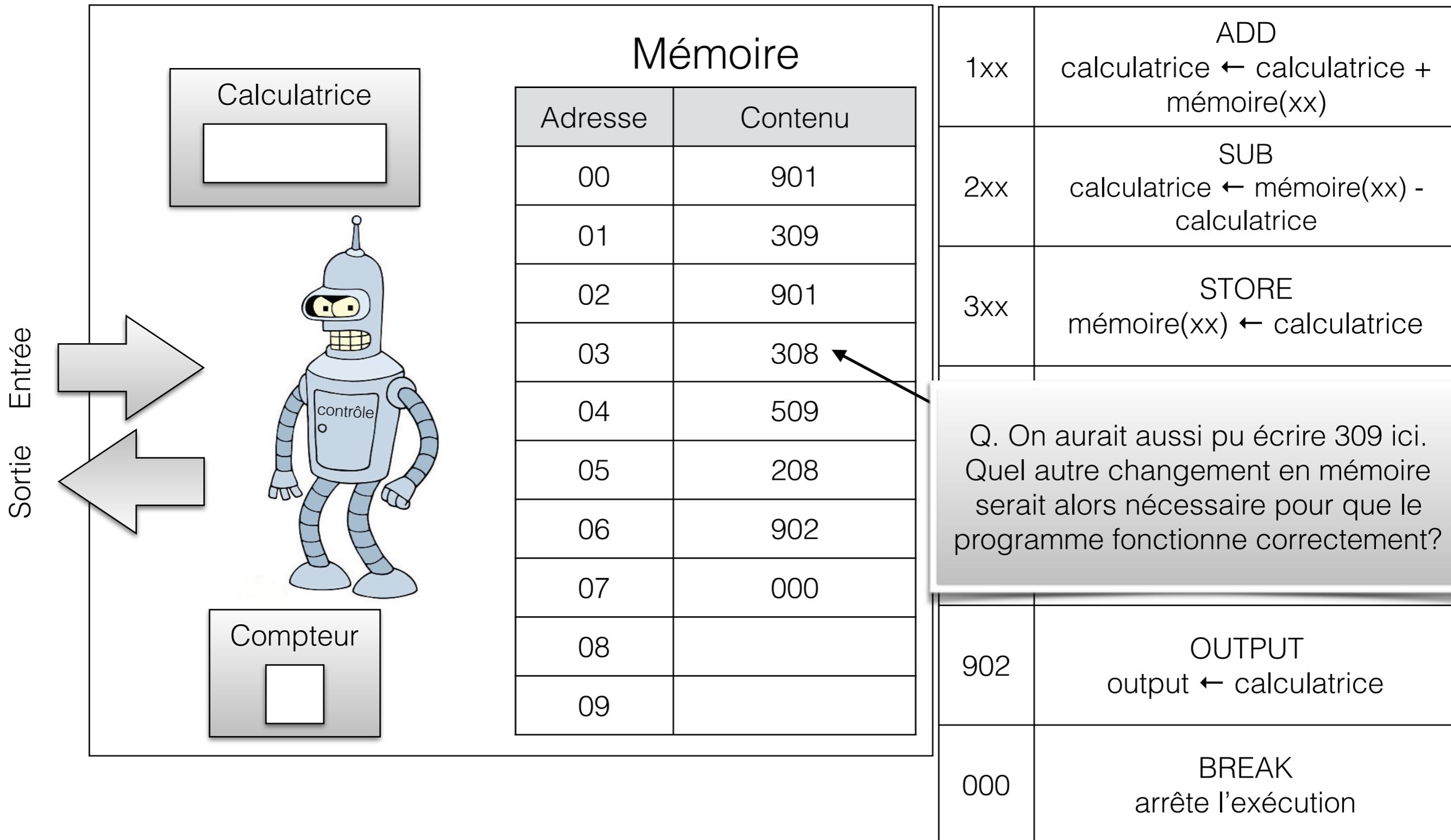
# Programme: sortie = entrée2 - entrée1?

Instructions (3 chiffres)



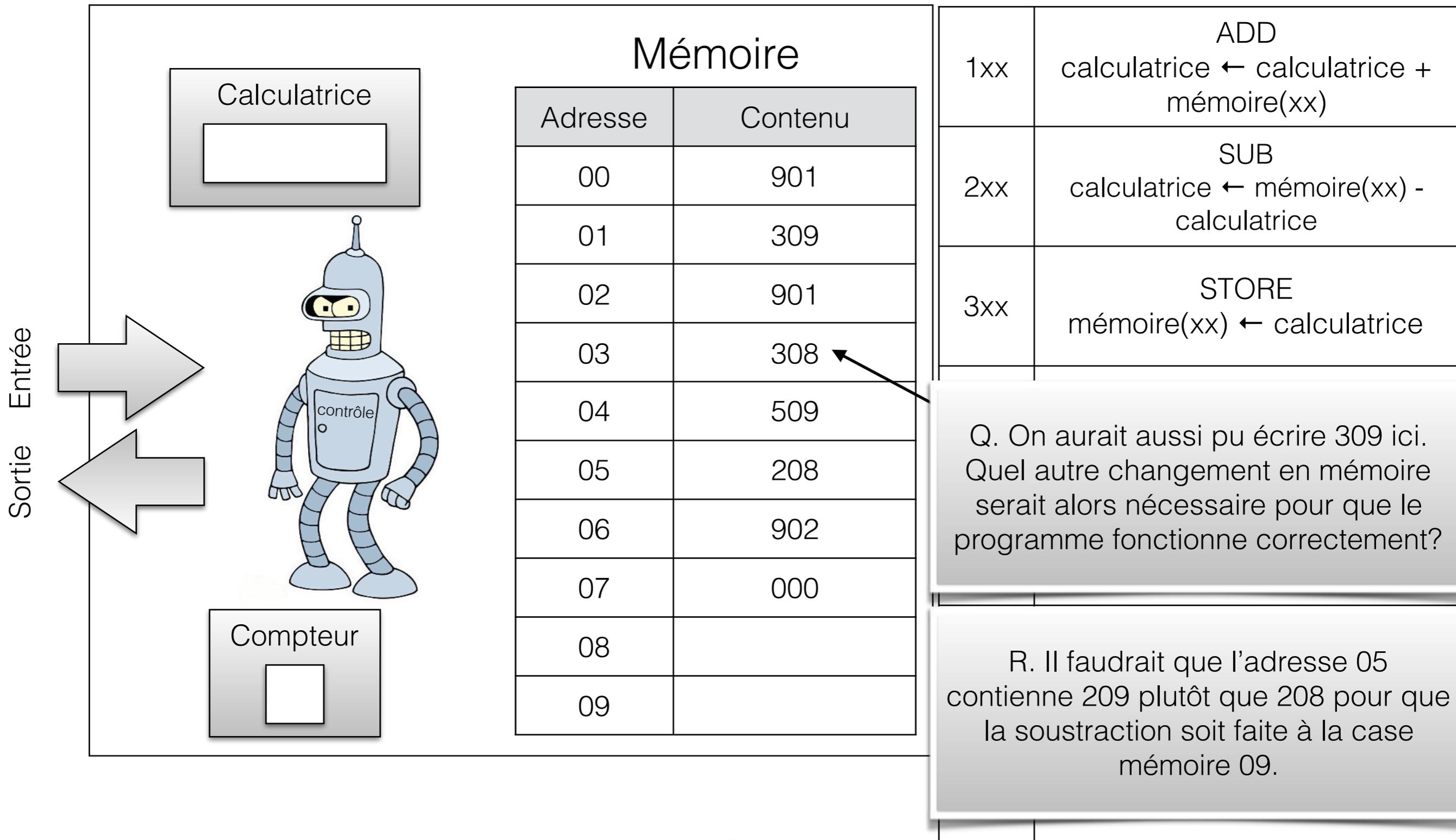
# Programme: sortie = entrée2 - entrée1?

Instructions (3 chiffres)



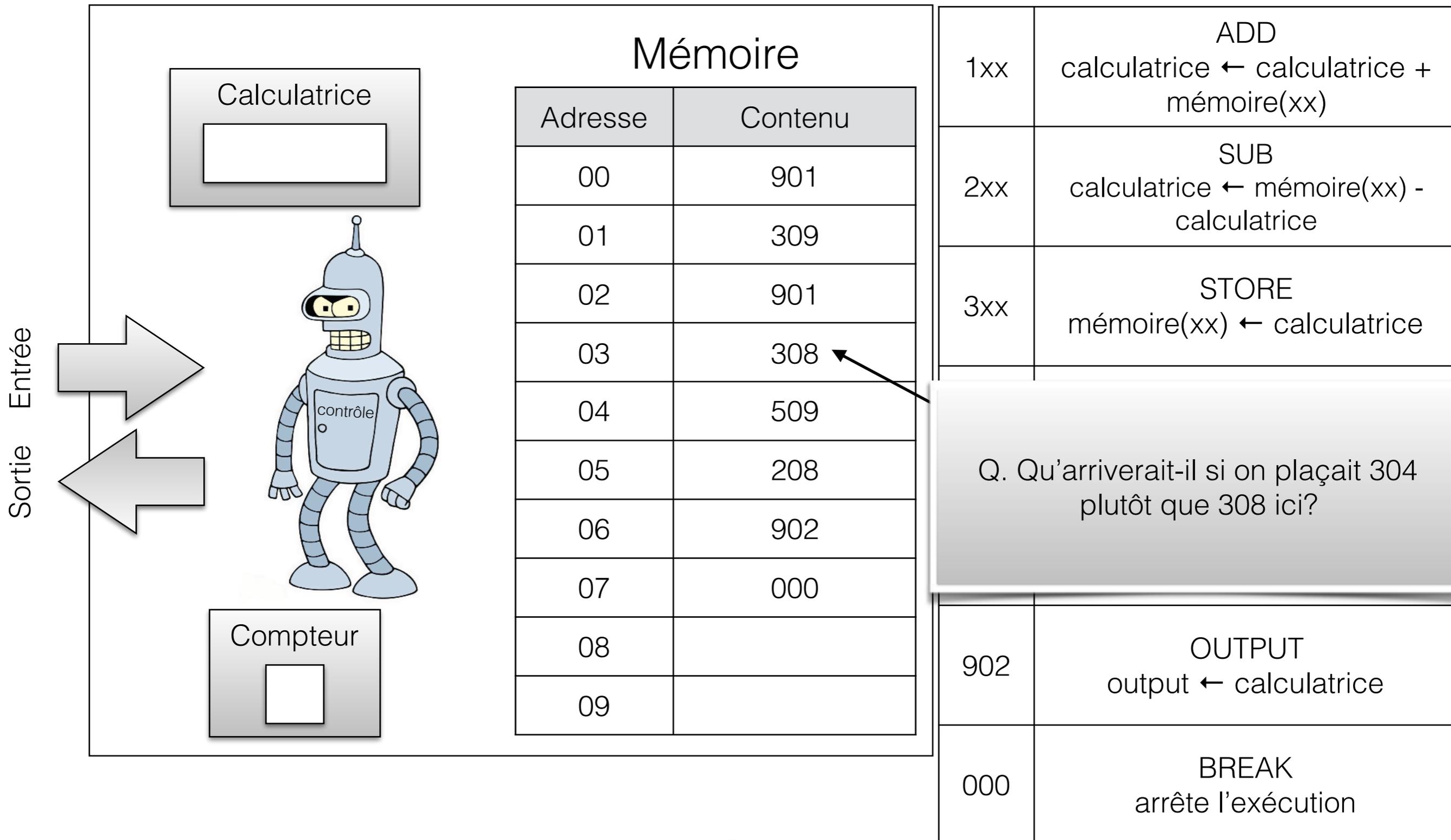
# Programme: sortie = entrée2 - entrée1?

Instructions (3 chiffres)



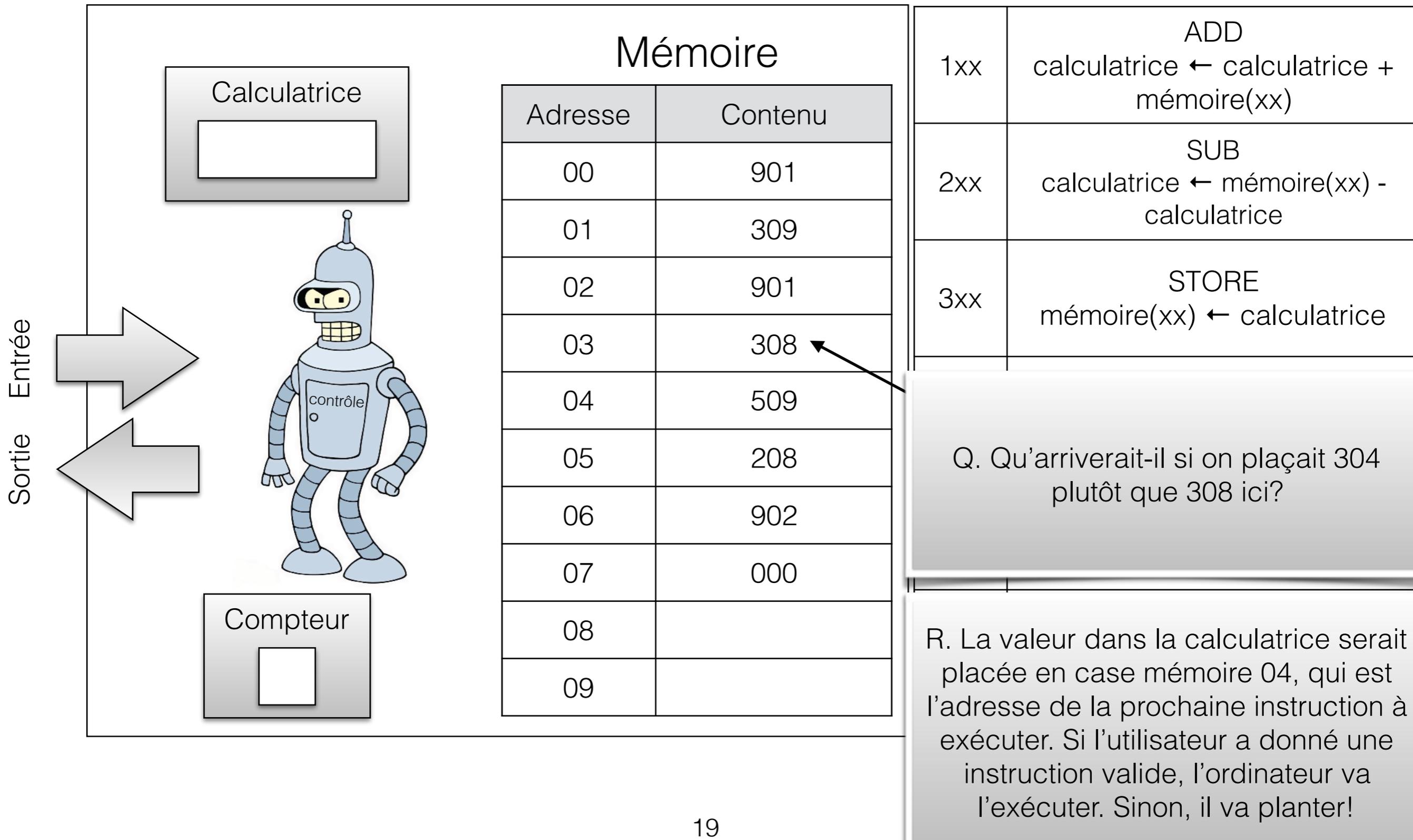
# Programme: sortie = entrée2 - entrée1?

Instructions (3 chiffres)



# Programme: sortie = entrée2 - entrée1?

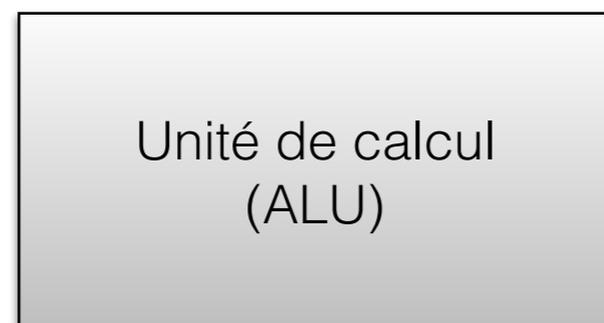
Instructions (3 chiffres)



**Quelles sont les composantes d'un microprocesseur?**

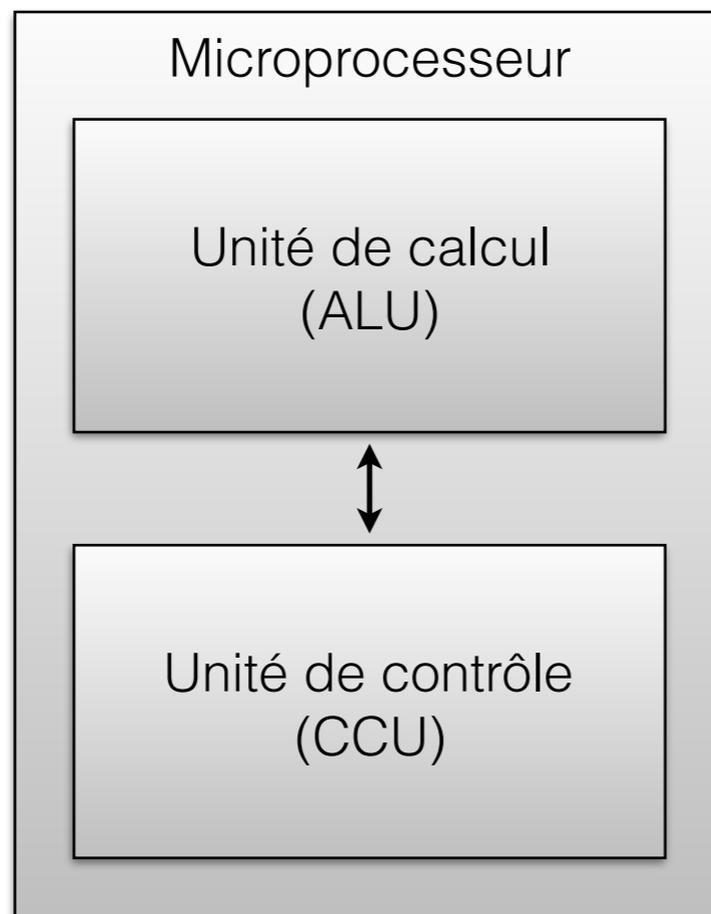
# Architecture von Neumann

- Unité de calcul
  - arithmétique: addition, soustraction, multiplications, etc.
  - logique: opérateurs logiques, comparaisons, etc.
- ALU (Arithmetic and Logic Unit)



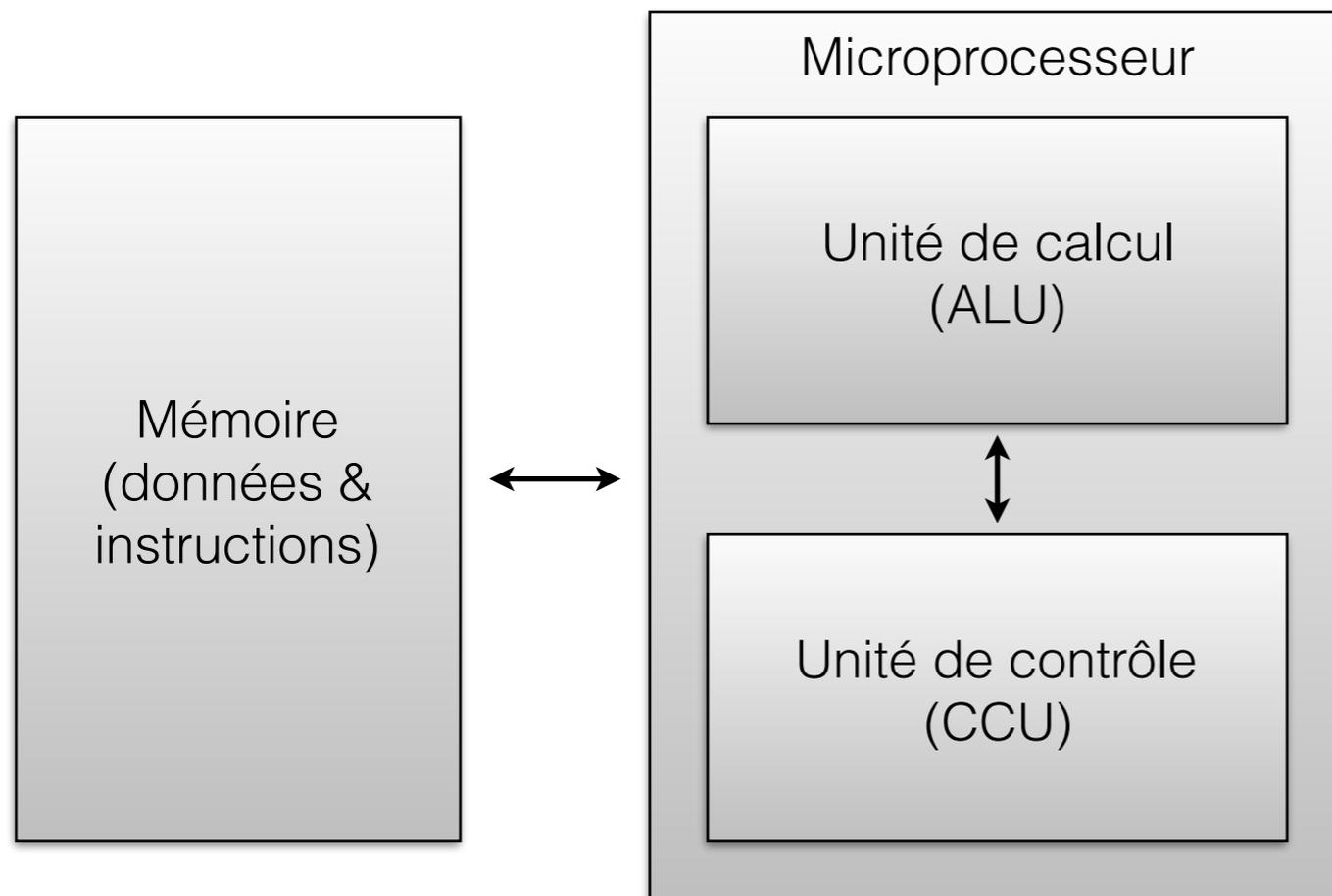
# Architecture von Neumann

- Unité de contrôle
  - Le coeur de l'ordinateur: va chercher des instructions, et les exécute
  - Se sert de l'ALU pour les tâches arithmétiques et logiques
- CCU (Central Control Unit)



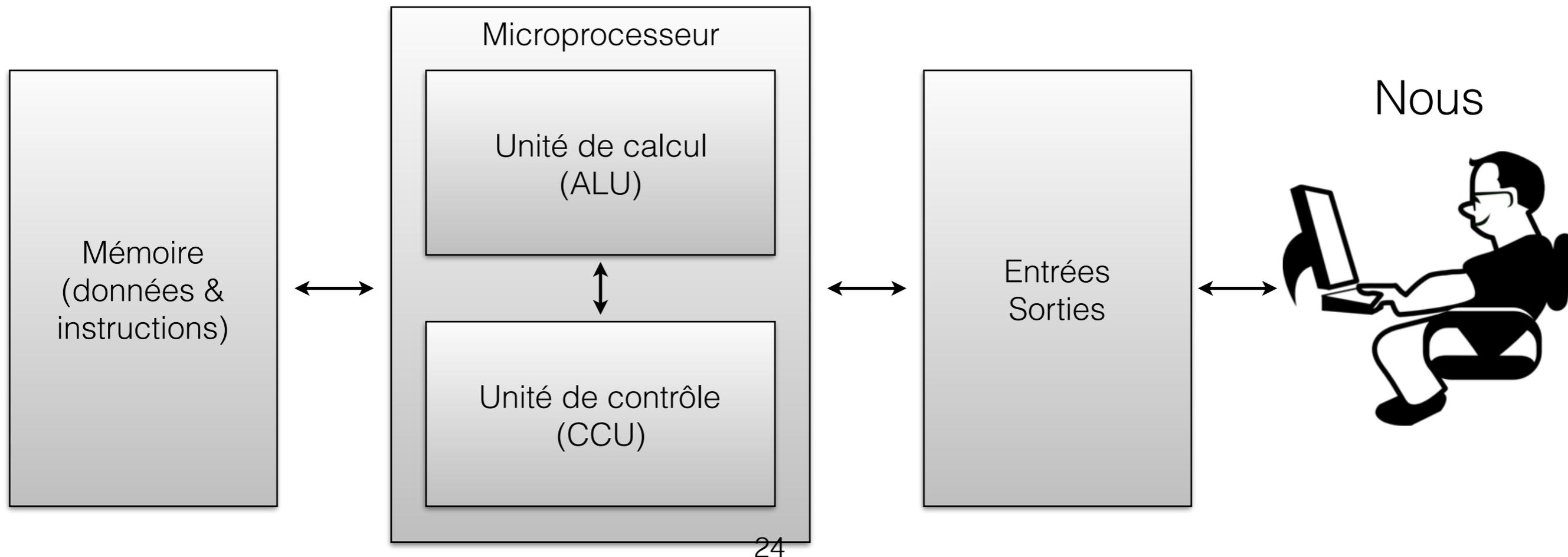
# Architecture von Neumann

- Mémoire
  - stocke les données manipulées par le microprocesseur
  - stocke les instructions (programmes) à exécuter par le microprocesseur
- Plusieurs “niveaux” de mémoire



# Architecture von Neumann

- Périphériques d'entrée-sorties
  - Nous permettent d'interagir avec l'ordinateur!
    - entrées: clavier, souris, lecteur DVD, etc.
    - sorties: écran, carte graphique, imprimante, etc.
    - entrées-sorties: disque dur, port ethernet, etc.



# Cycle d'instructions

- Que fait le microprocesseur?
  - Lis (*fetch*): va chercher la prochaine instruction
  - Décode (*decode*): détermine ce qu'il doit faire, à partir de l'instruction
  - Exécute (*execute*): exécute l'instruction

