

# Ordinateur (papier) Cardiac



Le but de cette fiche est la découverte du langage machine à travers un système particulier qui modélise le fonctionnement n'importe quel ordinateur actuel, l'ordinateur Cardiac. Il en existe une version papier, qui n'est pas vraiment un ordinateur puisque lorsqu'on l'utilise, c'est à nous de réaliser une partie des opérations (dont une partie du contrôle et les calculs), mais les actions s'y enchaînent comme sur un ordinateur réel.

Nous utiliserons une version en ligne de cet ordinateur, disponible [ici](#).

Le principe, pour le faire fonctionner dans le premier exercice, sera le suivant :

1. Copier dans la mémoire et/ou le ruban d'entrée les informations initiales souhaitées.
2. Mettre la valeur du compteur ordinal comme précisé.
3. Exécuter pas à pas les instructions et bien réfléchir au fonctionnement à chaque étape. Pour ce faire, on pourra regarder le fichier contenant le patron du modèle papier, qui indique l'ordre des actions à effectuer à chaque pas, puis plus tard la documentation du Cardiac.

## Exercice 1

Initialiser tout d'abord l'ordinateur en suivant les instructions suivantes :

- Sur le ruban d'entrée, placer les valeurs suivantes dans cet ordre :

023

010

- Sur la mémoire centrale, placer le contenu suivant aux numéros de cellules indiqués :

20. 034

23. 635

26. 136

21. 035

24. 628

27. 900

22. 234

25. 436

28. 009

- Enfin, mettre le compteur ordinal sur la valeur 20.

1. Faire fonctionner le Cardiac pas à pas jusqu'à l'arrêt de l'ordinateur. Que fait-il ?
2. Pouvait-on prévoir avant l'exécution si les cellules mémoire entre les adresses 20 et 28 contenaient des instructions ou des données ?
3. Dans ce programme, où sont les instructions et où sont les données ?

## Exercice 2

Écrire en **langage machine**, c'est à dire sous la forme de nombres décimaux sur trois caractères devant se trouver à certaines adresses mémoire, un programme Cardiac lisant un nombre sur le ruban d'entrée et imprimant 1 sur le ruban de sortie si ce nombre est supérieur ou égal à 10 ou 0 s'il est strictement inférieur. Pour cela, il faudra utiliser le code d'opération 8 (JAZ). Tester le programme après avoir pensé à mettre une valeur sur le ruban d'entrée et la bonne valeur initiale du compteur ordinal (autrement dit le numéro de cellule du début du programme).

Plutôt que d'utiliser le langage machine, qui n'est pas très parlant pour les humains (à moins peut-être d'y être plongé depuis des années), on utilise un langage intermédiaire, le **langage assembleur**. Pour cela, on donne un nom aux instructions du processeur (voir le jeu d'instructions sur la partie principale du processeur Cardiac). On peut également nommer des lignes (donc en particulier utiliser des noms de variables). Une traduction en langage assembleur du programme exécuté au tout début est :

```
.at 20
INP 34
INP 35
LDA 34
ADD 35
ADD n
STA 36
OUT 36
HRS 00
n: .word 009
```

ou, en nommant les espaces mémoire utilisés (et en les initialisant, ici à 000) :

```
.at 20
INP a
INP b
LDA a
ADD b
ADD n
STA x
OUT x
HRS 00
n: .word 009

.at 34
a: .word 000
b: .word 000
x: .word 000
```

Voir le fichier de documentation de Cardiac pour plus d'informations.

### Exercice 3

Traduire en langage assembleur le programme écrit à l'exercice 2.  
Tester le programme en le chargeant sur le simulateur.

### Exercice 4 (Facultatif)

Écrire un programme en assembleur Cardiac lisant un nombre *n* sur le ruban d'entrée et écrivant la suite des nombres de 1 à *n* sur le ruban de sortie.

### Exercice 5 (Facultatif)

Écrire un programme en assembleur Cardiac calculant la multiplication de deux nombres lus sur l'entrée et écrivant le résultat sur la sortie.

Python dispose d'une bibliothèque permettant de traduire un code Python en langage assembleur de l'ordinateur (qui n'est pas le même que le langage Cardiac, bien sûr). Cette bibliothèque est appelée *dis*.

**Exemple** Copier et exécuter le code Python suivant (attention à bien indenter) :

```
import dis

code = """
x=3
if x<0:
    y=-x
else:
    y=x
"""

dis.dis(code)
```