

Devoir sur table n° 1

Question 1

barème : 2 pt

Décrire brièvement le cycle d'exécution du processeur dans une machine de Von Neumann.

Question 2

barème : 2 pt

Décrire brièvement le cycle d'exécution d'un système d'exploitation mono-tâche.

Question 3

barème : 3 pt

Les questions suivantes concernent une machine de Von Neumann dont le jeu d'instructions est donné à la fin du sujet. Il s'agit du jeu d'instructions du simulateur AMIL.

1. Décrire (en français) un algorithme en trois étapes qui échange les contenus des mémoires 100 et 101.
2. Ecrire le programme correspondant en utilisant le jeu d'instructions d'AMIL. Commenter le programme afin de le rendre compréhensible.

Question 4

barème : 5 pt

On considère un appartement équipé d'un chauffage électrique, et d'un abonnement avec l'option *Heures Pleines / Heures Creuses*, qui permet de payer moins chère l'électricité la nuit. Par soucis d'économie, on souhaite que le chauffage se mette en route automatiquement pendant les heures creuses et s'éteigne pendant les heures pleines.

On suppose que le début des heures pleines est inscrit dans la mémoire à l'adresse 100, que le début des heures creuses se trouve à l'adresse 101 et que l'heure actuelle se trouve à l'adresse 102 (les heures sont des entiers compris entre 0 et 24). On souhaite que la machine AMIL décide de l'état du chauffage, et écrive à l'adresse 103 la valeur 1 s'il faut que les radiateurs soient (ou restent) allumés, ou bien la valeur 0 s'il doivent être (ou rester) éteints.

1. Décrire (en français) un algorithme répondant à ce souhait.
2. Ecrire et commenter le programme correspondant en utilisant le jeu d'instructions d'AMIL.
3. Faire les traces d'exécution du programme :
 - (a) lorsque les cases mémoires 100, 101 et 102 contiennent respectivement les valeurs 7, 22, et 1 ;
 - (b) lorsque les cases mémoires 100, 101 et 102 contiennent respectivement les valeurs 7, 22 et 12 ;
 - (c) lorsque les cases mémoires 100, 101 et 102 contiennent respectivement les valeurs 7, 22 et 23 !

Question 5

barème : 4 pt

On appelle n l'entier stocké à l'adresse 100 de la mémoire.

1. Décrire en français l'algorithme permettant de placer $(n!)$ à l'adresse 101 de la mémoire.
2. Ecrire et commenter le programme correspondant en utilisant le jeu d'instructions d'AMIL.
3. Faire la trace d'exécution du programme lorsque n vaut 3.

Question 6

barème : 4 pt

Soient deux vecteurs de dimension n , dont les composantes sont des entiers relatifs :

$$\vec{u} = \begin{pmatrix} u_1 \\ u_2 \\ \dots \\ u_n \end{pmatrix}, \vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ \dots \\ v_n \end{pmatrix} \text{ avec } u_1 \dots u_n, v_1 \dots v_n \in \mathbb{Z}.$$

Le but de cet exercice est de faire calculer leur produit scalaire ($\vec{u} \cdot \vec{v} = \sum_{i=1}^n u_i v_i$) par le simulateur AMIL.

On suppose que la dimension n des deux vecteurs est stockée à l'adresse 100 de la mémoire.

Les composantes u_1, u_2, \dots, u_n de \vec{u} sont stockées aux adresses 101, 102, \dots , $100 + n$.

Les composantes v_1, v_2, \dots, v_n de \vec{v} sont stockées aux adresses $100 + n + 1$, $100 + n + 2$, \dots , $100 + 2n$.

1. Décrire en français un algorithme qui calcule le produit scalaire $\vec{u} \cdot \vec{v}$ et qui stocke le résultat à l'adresse 99 de la mémoire.
2. Ecrire le programme correspondant en utilisant le jeu d'instructions d'AMIL. Commenter le programme autant que possible. Préciser en particulier le rôle de chacun des registres que le programme utilise.

Rappel du jeu d'instructions du simulateur AMIL

stop	Arrête l'exécution du programme.
noop	N'effectue aucune opération.
saut i	Met le compteur ordinal à la valeur i .
sisaut ri j	Si la valeur contenue dans le registre i est positive ou nulle, met le compteur ordinal à la valeur j .
init x ri	Initialise le registre i avec la valeur x .
lecture i rj	Charge, dans le registre j , le contenu de la mémoire d'adresse i .
lecture *ri rj	Charge, dans le registre j , le contenu de la mémoire dont l'adresse est la valeur du registre i .
ecriture ri j	Écrit le contenu du registre i dans la mémoire d'adresse j .
ecriture ri *rj	Écrit le contenu du registre i dans la mémoire dont l'adresse est la valeur du registre j .
inverse ri	Inverse le signe du contenu du registre i .
add x rj	Ajoute x au contenu du registre j .
add ri rj	Ajoute la valeur du registre i à celle du registre j .
mult, div, et	Même syntaxe que pour add mais pour la multiplication, la division entière et le et bit à bit.