Question 1:

On représente chaque état par un label suivi du traitement qui correspond a toutes les transitions sortantes de cet état.
Ainsi, en utilisant l'instruction JUMP, on pourrait facilement passer d'un état à un autre (sans utiliser de pile).

Les états finaux placeront leurs identifiants dans le reg R2.
Puis avant l'arrêt de la machine virtuelle, on récupère le contenu de ce reg pour l'écrire dans R0.

```
; ---------------------
(JMP <etat_INIT>)
; ---------------------
(LABEL E0)
; ---------------------
(CAR R0 R1) ; sauvegarder le symbole à comparé.
(CDR R0 R0) ; faire avancer le mot
; ---------------------
(CMP R1 a) ; faire la comparaison avec la transition 'a'
(JEQ E1) ; saut vers l'état de label 1
; ---------------------
(CMP R1 b) ; faire la comparaison avec la transition 'b'
(JEQ E0) ; saut vers l'état de label 0
; ---------------------
; symbole non reconnu alors on s'arrête
(MOVE NIL R0)
(HALT) ; ce qui suit un HALT / JMP / RET est forcément un LABEL
; ---------------------
;---------------------------------
(LABEL E3) ; Etat 3
; ---------------------
(MOVE E3 R2) ; sauvegarde de l'identifiant
(cmp R0) ; est ce qu'il reste un symbole à comparer, oui/non ?
(bnull OK) ; si oui, jump sur OK (move + halt)
; ---------------------
(CAR R0 R1) ; sauvegarder le symbole à comparé.
(CDR R0 R0) ; faire avancer le mot
; ---------------------
(CMP R1 a) ; faire la comparaison avec la transition 'a'
(JEQ E2) ; saut vers l'état de label 1
(CMP R1 b) ; faire la comparaison avec la transition 'b'
(JEQ E3) ; saut vers l'état 3
; ---------------------
; symbole non reconnu alors on s'arrête, on met nil dans le registre d'arrêt R0
(MOVE NIL R0)
(HALT) ; ce qui suit un HALT / JMP / RET est forcément un LABEL
;---------------------------------
(LABEL OK) ; le mot a bien été par l'automate
(MOVE R2 R0)
(HALT)
;---------------------------------
```

Question 2.1:

Notre fonction auto2vm concatène le résultat de trois fonctions auxiliaires.

La première permet de se brancher inconditionnellement sur l'état initial en ré-utilisant la fonction auto-init.

La seconde permet de réaliser le traitement de chaque état grâce à l'application d'une fonction map qui permet de réaliser une fonction (vm_etat_callback) sur chaque état via auto-etat-liste.

La fonction vm_etat_callback possède une lambda avec l'état en paramètre. Cette fonction lambda concatène les instructions d'un état final si nécessaire sinon simplement, celui de l'avancement du mot a traité ainsi que l'application d'une fonction map qui applique la fonction vm_transition_callback sur les transitions de cet état via auto-trans-list. Elle concatène aussi le jeu d'instructions qui permet dans tous les cas où le symbole n'est pas reconnu déplacer NIL dans R0.

La troisième retourne la suite d'instructions qui termine la VM en déplace l'identifiant de l'état final atteint.

```
;-------------------------------
les opérations utilisées pour lever les ambiguités :
HALT  ; arrêt
LABEL ; étiquete
CMP   ; comparaison
MOVE  ;déplacement registre à registre
JEQ   ; saut si égale
```