

## Manuel Utilisateur – Interpréteur Assembleur INF404

### Langage accepté

Notre interpréteur prend en charge un langage assembleur simplifié, composé de :

#### Instructions de base :

- MOV R1, #5 : met la valeur 5 dans R1
- ADD R1, R2, #1 :  $R1 \leftarrow R2 + 1$
- SUB, CMP, LSL, LSR, AND, ORR, TST

#### Branchements :

- BEQ label : saut si égalité
- BNE, BLT, BGT, BAL, BL

#### Mémoire :

- LDR R1, [R2] ou LDR R1, adresse
- STR R1, [R2]
- AFFRC R1 : afficher un caractère
- AFFAC R1 : afficher une chaîne (adresse en R1)
- AFFRN R1 : afficher un entier

#### Sections spéciales :

- .data → Définition de variables
- .ascii "abc" → Chaîne de caractères
- .dword 5 → Valeur entière
- .skip n → Réserve n cases mémoire

Test lexeme

```
ASM mov.s
.data
.text

mov r2,#1000
mov r4,r2

bal exit
```

```
racimz@PC-Racim:~/INF404/INF404/INF404-master$ ./test_lexeme test/mov.s
(ligne 1, colonne 1) : [nature = .DATA, chaine = .data]
(ligne 2, colonne 0) : [nature = RETOUR_LIGNE, chaine = ]
(ligne 2, colonne 1) : [nature = .TEXT, chaine = .text]
(ligne 3, colonne 0) : [nature = RETOUR_LIGNE, chaine = ]
(ligne 4, colonne 0) : [nature = RETOUR_LIGNE, chaine = ]
(ligne 4, colonne 1) : [nature = MOV, chaine = mov]
(ligne 4, colonne 5) : [nature = REG, chaine = r2, numéro registre = 2]
(ligne 4, colonne 7) : [nature = VIRGULE, chaine = ,]
(ligne 4, colonne 8) : [nature = ENTIER, chaine = #1000, valeur = 1000]
(ligne 5, colonne 0) : [nature = RETOUR_LIGNE, chaine = ]
(ligne 5, colonne 1) : [nature = MOV, chaine = mov]
(ligne 5, colonne 5) : [nature = REG, chaine = r4, numéro registre = 4]
(ligne 5, colonne 7) : [nature = VIRGULE, chaine = ,]
(ligne 5, colonne 8) : [nature = REG, chaine = r2, numéro registre = 2]
(ligne 6, colonne 0) : [nature = RETOUR_LIGNE, chaine = ]
(ligne 7, colonne 0) : [nature = RETOUR_LIGNE, chaine = ]
(ligne 7, colonne 1) : [nature = BAL, chaine = bal]
(ligne 7, colonne 5) : [nature = EXIT, chaine = exit]
(ligne 8, colonne 0) : [nature = RETOUR_LIGNE, chaine = ]
(ligne 9, colonne 0) : [nature = RETOUR_LIGNE, chaine = ]
(ligne 10, colonne 0) : [nature = RETOUR_LIGNE, chaine = ]
```

Test syntaxe

```
ASM beq_faux.s
.data
.text

mov r1,#10
cmp r1,#2
beq vrai
faux:
mov r2,#99
affrn r2
bal fin_si
vrai:
mov r3,#1
fin_si:
bal exit
```

```
racimz@PC-Racim:~/INF404/INF404/INF404-master$ ./test_syntaxe test/beq_faux.s
Analyse syntaxique en cours...
99
===== [ZONE DE DÉBOGAGE] =====
Affichage de l'arbre AST

[1] MOV
  Registre: R1
  Valeur: 10
[2] CMP
  Registre: R1
  Valeur: 2
[3] BEQ
  Étiquette: vrai
[4] MOV
  Registre: R2
  Valeur: 99
[5] AFFRN
  Registre: R2
[6] BAL
  Étiquette: fin_si
[7] MOV
  Registre: R3
  Valeur: 1
[8] FIN
```

État des Registres			
R0 =	0	R1 =	10
R2 =	99	R3 =	0
R4 =	0	R5 =	0
R6 =	0	R7 =	0
R8 =	0	R9 =	0
R10 =	0	FP =	1000
R12 =	0	SP =	1000
LR =	0	PC =	0

```
===== [FIN DE DÉBOGAGE] =====
Analyse réussie
```