

TD07 - CODE TROIS ADRESSES

Pour chacun des programmes des exercices suivants, dessiner l'arbre abstrait et donner le code trois adresses correspondants.

1. EXPRESSION ARITHMÉTIQUE

```
1 entier result;
2 main()
3 {
4     result = 1 * (2 + 3) - (4 * (5 + 6));
5 }
```

2. INSTRUCTION si-sinon ET COMPARAISON

```
1 entier a;
2 main() {
3     a = lire();
4     si a < 0 alors{
5         ecrire(0);
6     }
7     sinon{
8         ecrire(a);
9     }
10 }
```

3. BOUCLE ET OPÉRATIONS LOGIQUES

```
1 entier n;
2 main() {
3     tantque n < 100 & !(n < 0) faire {
4         n = lire();
5     }
6 }
```

4. APPELS DE FONCTION

```
1 discriminant(entier a, entier b, entier c){
2     retour(b * b - 4 * a * c);
3 }
4 main (){
5     ecrire(discriminant(3,4,5));
6 }
```

5. TABLEAUX

```
1 entier tab[10];
2 main()
3 entier i;
4 {
5     i = 1;
6     tab[0] = 1;
7     tantque i < 10 faire {
8         tab[i] = tab[i - 1] + 1;
9         i = i + 1;
10    }
11 }
```

6. SPÉCIFICATION DU CODE TROIS ADRESSES

Le tableau ci-dessous décrit les instructions autorisées dans le code trois adresses. En plus de ces instructions, vous pouvez ajouter des étiquettes (**e** :) à certaines lignes. Les adresses utilisées peuvent être des constantes entières (**c**), des valeurs temporaires créés pour évaluer des expressions complexes (**t**), des variables du programme source en L (**v**) ou des étiquettes (**e**). Le tableau indique, pour chaque type d'opération, le nombre et les types des trois adresses **a1**, **a2** et **r**. Par exemple, la première adresse **a1** d'un saut direct est une étiquette **e** alors que la troisième adresse **r** d'une opération arithmétique peut être une valeur temporaire ou une variable (**tv**).

Type	Opérations	a1	a2	r	Syntaxe
arithmétique	+ - * /	ctv	ctv	tv	r = a1 op a2
affectation	=	ctv		tv	r = a1
saut test	== < <= > >=	ctv	ctv	e	if a1 op a2 goto r
saut direct	goto	e			goto a1
appel fonction	call	e		tv	r = call a1 ou call a1 ¹
lecture	read			tv	r = read
écriture	write	ctv			write a1
e/s fonction	param ret	ctv			op a1
début/fin fonc.	fbegin fend				fbegin, fend