

**Série n°4**  
**Compil**  
**Organisation des données à l'exécution**

**Exercice 1:**

Soit le programme Fortran suivant :

```
Program
Integer A, B, C, F, G, K, N
Integer D(5), E(5)
Common 1 A, G
Common 2 K, N
Equivalence G, D(2)
Equivalence D(4), E(1)
....
Call Sub(B, F)
Call Mult(F,G)
Stop
End.
```

```
Subroutine Sub(x,y)
Integer H, I, B(5), L(3), J
Common 1 H, I
Equivalence B(3), J
Equivalence L(1), B(4)
....
Return
End.
```

```
Subroutine Mult(x,y)
Integer M, T(2), S
Common 2 M, S
....
Return
End.
```

Donner les zones de données du PP, du SP et du Common.

[Solution](#)

---

**Exercice 2:**

Soit le programme suivant :

```
Begin
  Real A, E;
  Integer array B1[1:A];
  Integer K;
  L1: Begin
    Integer array B2[1:E];
    Integer I;
    Procedure X(C, D);
    Real C,D;
    Begin
      Array B3[1:K; 1:2*K];
      L3: B3[I,1]:= B1[I]+ A;
```

```

End;
L2: B1[I] := I - A;
End;
A := E + K;
End;

```

- Donner le contenu de la pile de données aux instructions L1, L2 et L3.
- Calculer les adresses absolues des variables de L3.

### Solution

## Exercice 3: Optimisation du code

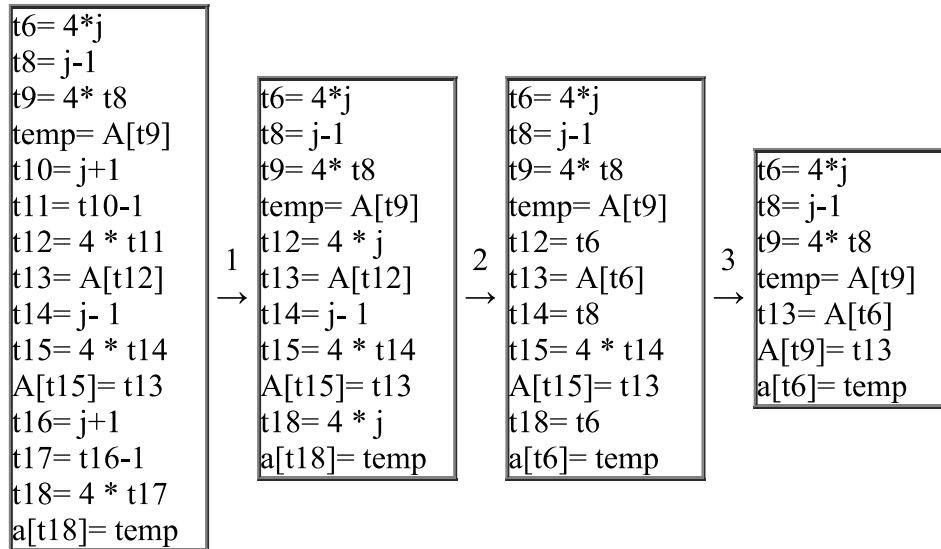


Figure 1: Optimisations successives

On considère les transformations successives appliquées au code assembleur de la figure 1. On considère que l'on a appliqué des optimisations qui sont parmi les suivantes :

- Propagation de copie (e.g. remplacer  $t1=t2$ ;  $t3=4*t1$  par  $t1=t2$ ;  $t3=4*t2$ ).
- Propagation d'expression (e.g. remplacer  $t1=expr$ ;  $t3=4*t1$  par  $t1=expr$ ;  $t3=4*expr$ ).
- Elimination d'expressions redondantes (communes).
- Elimination de code inutile.
- Simplification algébrique (e.g. remplacer  $t1+1-1$  par  $t1$ ) associée à une autre transformation.

1. Indiquez pour les étapes : 1, 2 et 3 de la figure 1 quelles optimisations ont été effectuées. Fournir le code des étapes intermédiaires.

2. Appliquer maintenant une séquence de ces transformations sur le code de la figure 2 ci-dessous de façon à minimiser la taille du code résultant.

```

t1= t0
t2= 4
t3= t1 *t1
t4= t2 + 2
t5= t0 ^ 2 // t0 puissance 2
t6= t5+ t3
t7= t4 * t6

```

Figure 2. Code à optimiser

[Solution](#)

---

**Exercice 4:**

Eliminer les sous expressions communes dans la portion de programme suivante :

1.  $y := 2 * a$
2.  $a := x + y$
3.  $x := a + b$
4.  $y := i + 1$
5. si  $b < 10$  Aller à (8)
6.  $z := x + y$
7. Aller à (14)
8.  $d := i + 1$
9.  $y := a + b$
10.  $a := 2 * z$
11.  $z := a + b$
12.  $i := 2 * i$
13. si  $z < 50$  Aller à (18)
14.  $e := x + y$
15.  $x := a + b$
16.  $d := z * d$
17. Aller à (8)
18.  $x := 2 * a$
19.  $i := i + 1$
20. si  $x < 100$  Aller à (14)
21.  $b := x + y$
22.  $c := 2 * z$
23.  $d := a + b$

[Solution](#)