

BLACK 03/01/12

Examen de compilation2

Exercice 1 (5 points) :

Soit la matrice creuse suivante d'ordre n, n est impaire.

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 & \dots & 1 & 1 & 1 & 1 \\ & 1 & & 0 & & & & 1 & \\ & & 1 & & & & & & 1 \\ & & & & \dots & & & & \\ 0 & & & 1 & & & & & 0 \\ & & & & 1 & & 1 & & \\ & 1 & & 0 & & & & 1 & \\ 1 & 1 & 1 & 1 & \dots & 1 & 1 & 1 & 1 \end{pmatrix}$$

C'est à dire les éléments des deux diagonales sont différents de zéro, ainsi que la première et la dernière ligne, les autres éléments sont nuls.

On veut représenter dans une zone contiguë que les éléments non nuls de la matrice M. les éléments sont rangés ligne par ligne

- a- Trouver la relation qui relie les indices i et j des éléments non nuls.
- b- Donner l'adresse des éléments M[i,j] rangés ligne / ligne

Exercice 2 :

Partie I (4 points)

Soit les deux portions de programmes :

Programme1

DIMENSION A(4), B(6)
Common A
EQUIVALENCE (A(2), B(1))

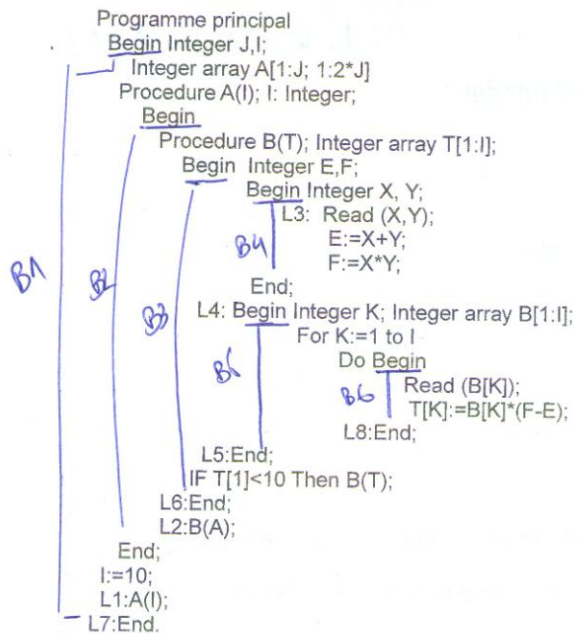
Programme2

DIMENSION A(4), B(6)
Common A
EQUIVALENCE (A(2), B(3))

- a) Quelle est la différence entre COMMON et EQUIVALENCE?
- b) Parmi les deux programmes, lequel est invalide? Pourquoi?

Partie II (6 points)

Soit le programme suivant :



a - Donner les différents états de la pile aux étiquettes L1, L2, L3, L4, L5, L6, L7 et L8.

b - Donner les adresses des variables de l'instruction $T[K] := B[K] * (F - E)$

c (Active Area)

Exercice 3 (5 points):

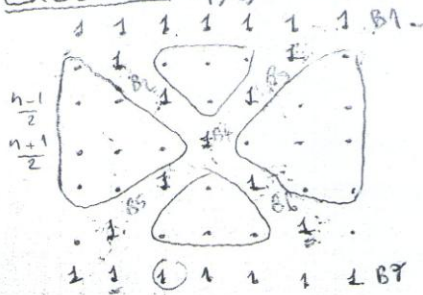
a) Éliminer les sous-expressions communes dans la portion de programme suivante:

- | | |
|-----------------------------|---------------------|
| 1- $i := m - 1$ | 16- $t7 := 4 * i$ |
| 2- $j := n$ | 17- $t8 := 4 * j$ |
| 3- $t1 := 4 * n$ | 18- $t9 := a[t8]$ |
| 4- $v := a[t1]$ | 19- $a[t7] := t9$ |
| 5- $i := i + 1$ | 20- $t10 := 4 * j$ |
| 6- $t2 := 4 * i$ | 21- $a[t10] := x$ |
| 7- $t3 := a[t2]$ | 22- aller à (5) |
| 8- Si $t3 < v$ aller à (5) | 23- $t11 := 4 * i$ |
| 9- $j := j - 1$ | 24- $x := a[t11]$ |
| 10- $t4 := 4 * j$ | 25- $t12 := 4 * i$ |
| 11- $t5 := a[t4]$ | 26- $t13 := 4 * n$ |
| 12- Si $t5 > v$ aller à (9) | 27- $t14 := a[t13]$ |
| 13- Si $i > j$ aller à (23) | 28- $a[t12] := t14$ |
| 14- $t6 := 4 * i$ | 29- $t15 := 4 * n$ |
| 15- $x := a[t6]$ | 30- $a[t15] := x$ |

b) Donner deux autres types d'optimisations en expliquant avec un exemple.

Correction de l'examen COMPILE 2 (2011/2012)

Exercice 1 5pts



i/ Les relations (2pts)

- $M[i, j] \in B1 \Rightarrow i = 1 \text{ et } 1 \leq j \leq n$
- $M[i, j] \in B2 \Rightarrow 2 \leq i \leq \frac{n-1}{2} \text{ et } i = j$
- $M[i, j] \in B3 \Rightarrow 2 \leq i \leq \frac{n-1}{2} \text{ et } j = n-i+1$
- $M[i, j] \in B4 \Rightarrow i = \frac{n+1}{2} \text{ et } j = \frac{n+1}{2}$
- $M[i, j] \in B5 \Rightarrow \frac{n+3}{2} \leq i \leq n-1 \text{ et } j = n-i+1$
- $M[i, j] \in B6 \Rightarrow \frac{n+3}{2} \leq i \leq n-1 \text{ et } j = i$
- $M[i, j] \in B7 \Rightarrow i = n \text{ et } 1 \leq j \leq n$

les adresses (3pts)

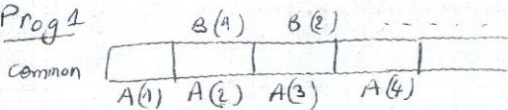
- $M[i, j] \in B1 \Rightarrow \text{ad } M[i, j] := \text{ad base} + (j-1) * \text{taille d'1 elt.}$
- $M[i, j] \in B2 \Rightarrow \text{ad } M[i, j] := \text{ad base} + \frac{n}{2} + e(i-2) * \text{taille d'1 elt.}$
- $M[i, j] \in B3 \Rightarrow \text{ad } M[i, j] := \text{ad base} + n + 2(i-2) + 1 * \text{taille d'1 elt.}$
- $M[i, j] \in B4 \Rightarrow \text{ad } M[i, j] := \text{ad base} + n + 2(i-2) * \text{taille d'1 elt.}$
- $M[i, j] \in B5 \Rightarrow \text{ad } M[i, j] := \text{ad base} + n + 2(i-2) * \text{taille d'1 elt.}$
- $M[i, j] \in B6 \Rightarrow \text{ad } M[i, j] := \text{ad base} + n + 2(n-2) + (i-1) * \text{taille d'1 elt.}$
- $M[i, j] \in B7 \Rightarrow \text{ad } M[i, j] := \text{ad base} + n + 2(n-2) + (i-1) * \text{taille d'1 elt.}$

Exercice 2 3pts

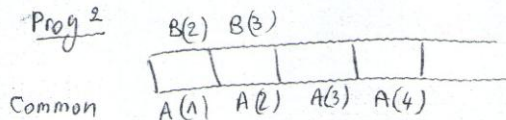
Partie I

- 1pt a/ COMMON partage une zone mémoire entre 2 pgms (LP et SP)
- EQUIVALENCE donne le m nom à 2 zones différentes dans un m pgm

b/ Prog 1



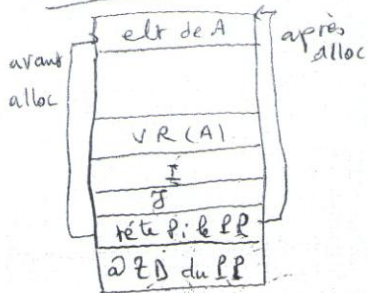
Prog 2



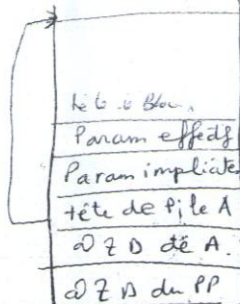
le programme 2 est invalide

car, on réserve d'abord l'espace pour le vecteur A(1) ensuite on exécute l'équivalence et dans ce cas (prog 2), on peut pas réserver pour B(1) car les elts d'1 vecteur doivent être dans une zone contigue.

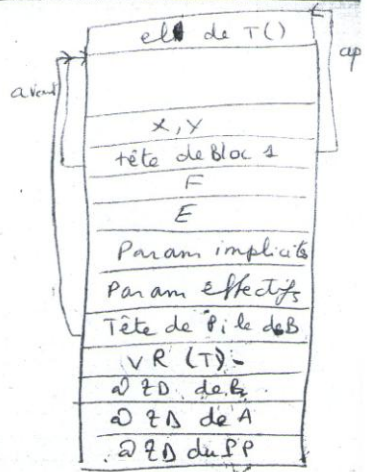
Partie II 6pt



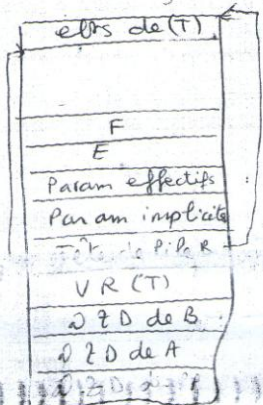
0,5 ZD du PP (L1)



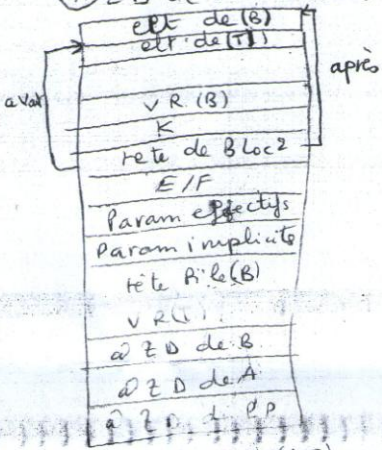
0,5 ZD de Proc A (L2)



ZD de Proc B (L3) ①

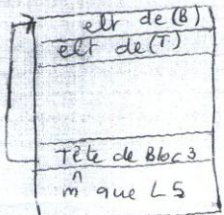


0,5 ZD de Proc B (L4)



ZD de Proc B (L5) ①

0,5 $L_6 = L_4$



0,5 ZD du Proc B (L8)

Les adresses 1pt
 $T[K] := B[K] * F - E$

- ① $K := c(\text{activearea}) + \text{deplacement de la variable } K / \text{debut de ZD de}$
- ② $F := c(\text{activearea}) + \text{ " " " F / " " ZD de}$
- ③ $E := \text{ " " " + " " " " E / " " de}$
- ④ $T[K] := \text{ " " + deplacement de } T[1] / \text{debut de ZD}$
 $+ (K-1) * \text{taille d'1 elt}$
- ⑤ $B[K] := \text{ " " + deplacement de } B[1] / \text{debut de ZD de B}$
 $+ (K-1) * \text{taille d'1 elt}$

Exo3: 6 pts

corrigé examen compil 2

2011/2012

1 - $i := m - 1$

2 - $j := n$

3 - $t_1 := 4 * n$

4 - $v := a[t_1]$

5 - $i := i + 1$

6 - $t_2 := 4 * i$

7 - $t_3 := a[t_2]$

8 - Si $t_3 < n$ aller à (5)

9 - $j := j - 1$

10 - $t_4 := 4 * j$

11 - $t_5 := a[t_4]$

12 - Si $t_5 > n$ aller à (9)

13 - Si $i > j$ aller à (23) (21) (0.8)

14 - $t_6 := 4 * i$ [on peut pas supprimer car i change dans l'instruction 6: Aller (5)] (0.5)

15 - $x := a[t_6]$

16 - $t_7 := 4 * j$ [on peut pas supprimer car j ne change pas ds l'inst 14 et 15 avant 16] (0.5)

17 - $t_8 := a[t_7]$ (0.5)

18 - $a[t_8] := t_9$

19 - $t_{10} := 4 * j$ [on peut supprimer] $\Rightarrow t_{10} = t_8$ (0.5)

20 - $a[t_{10}] := x$

21 - aller à (5)

22 - $t_{11} := 4 * i$ [on peut pas supprimer car dans (5) i change] (0.5)

23 - $x := a[t_{11}]$

24 - $t_{12} := 4 * i$ (0.5) $t_{12} = t_{11}$

25 - $t_{13} := x$ (0.5) $t_{13} = t_1$

26 - $t_{14} := a[t_{13}]$ (0.5) $t_{14} = x$

27 - $a[t_{14}] := t_{12}$ (0.5) $a[t_{14}] := x$

28 - $t_{15} := t_{14}$ (0.5) $t_{15} = t_{14}$

b / exple

(1 pt)