

Durée une 1h :30

1. Optimiser chaque portion de code et dites quel est le type d'optimisations utilisées?

Code 1
$$\begin{array}{l} x := y \\ s := x * f(x) \end{array}$$
Code2

```
a := b+c
c := b+c
d := b+c
```

Code3
$$\begin{array}{l} x := y + 1 \\ y := 5 \\ x := 2 * z \end{array}$$
Code4
$$c[i+1] := a[i+1] + b[i+1]$$
Code5

```
a:=5
if (a≠5) then x:=i+1
```

Code6

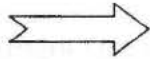
```
For(i=0;i<n;i++)
{ a[i]:=a[i]+c/y; }
```

2. Indiquez les optimisations effectuées pour le code suivant :

```

z := 1 ;
k := z - 1 ;
x := y * z + k ;

```


$$x := y$$

Exercise 2: (5 points)

Soit la matrice creuse d'ordre n , n pair.

$$A = \begin{pmatrix} 0 & X & 0 & X \\ X & 0 & X & 0 \end{pmatrix}$$

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

- a- Trouver la relation qui relie les indices i et j des éléments non nuls. (2pts)
b- Donner l'adresse des éléments $A[i,j]$ rangés colonne / colonne. (3pts)

Example :

$$A = \begin{pmatrix} \times & \times \times & 0 & 0 & \times \\ 0 & \times \times & 0 & \times \times \\ 0 & 0 & \times & \times \times \times \\ \times \times \times & \times & 0 & 0 \\ \times \times 0 & \times \times 0 \\ \times 0 0 & \times \times \times \end{pmatrix}$$

Exercice 3: (8 points)

Soit la portion du programme :

```
Procedure Proc A (x, y): integer x,y;  
L1: Begin  
    Reel i, j; integer array T1 [3:y]  
    L2: Procedure Proc B (c,d);integer c, d;  
        Begin  
            Integer Array T2[1 : x; x: x + y];  
            For k:= 1 to d  
                Do  
                    T2[k,c]:= T1 [k] * 2 ;  
                    L3: T1[k]:= T2[k,c]+x;  
                End;  
            Begin Integer a; Integer array T3[1:l];  
            L4: For a:=1 to l  
                Do Begin  
                    Read (T3[a]);  
                    T1[a]:=T3[a]*(x-y);  
                    L 6: a :=x+y*l/k-i*j  
                End;  
            L5:End;  
        End;  
    End;
```

- Donner le contenu de la pile de données aux instructions L 1, ..., L6. (3pts)
- Calculer les adresses absolues des variables de L3. (3pts)
- Générer le code objet de l'instruction L 6 en utilisant l'instruction GetInAcc (2pts)

Remarque :

Les réponses doivent être claires et complètes.

Bon courage.

Corrigé Examen compilation 07/01/2014

Exercice 1 (7pts)

1° Code 1

(1pt) $x := y$
 $S := x * f(x)$ $\xrightarrow[\text{de copie}]{\text{propagation}}$ $x := y$
 $S := y * f(y)$ $\xrightarrow[\text{code inutile}]{\text{elimination}}$ $S := y * f(y)$

Code 2

(1pt) $a := b + c$
 $c := b + c$
 $d := b + c$ $\xrightarrow[\text{expressions communes}]{\text{elimination pour}}$ $a := b + c$
 $c := d$ $\xrightarrow[\text{de copie}]{\text{propagation}}$ $a := b + c$
 $d := b + c$ $\xrightarrow[\text{de copie}]{\text{propagation}}$ $c := d$
 $d := b + a$

$\boxed{a := b + c}$
 $\boxed{d := b + a}$ $\xleftarrow[\text{elimination code mort}]{(2pt)}$

Code 3

(0,5pt) $x := y + 1$
 $y := 5$
 $x := 2 * 3$ $\xrightarrow[\text{code inutile}]{\text{elimination}}$ $x := 2 * 3$
 $y := 5$
 $x := 2 * 3$ $\xrightarrow[\text{code inutile}]{\text{elimination}}$ $\boxed{y := 5}$
 $x := 2 * 3$

Code 4

(1pt) $C[i+1] := A[i+1] + B[i+1]$ $\xrightarrow[\text{expressions communes}]{\text{elimination pour}}$ $\boxed{C[i] := A[i] + B[i]}$
 $C[x] := A[x] + B[x]$

Code 5

(1pt) $a := 5$
~~If~~ $a \neq 5$ Then $x := i + 1$ $\xrightarrow[\text{code inutile}]{\text{elimination}}$ $\boxed{a := 5}$

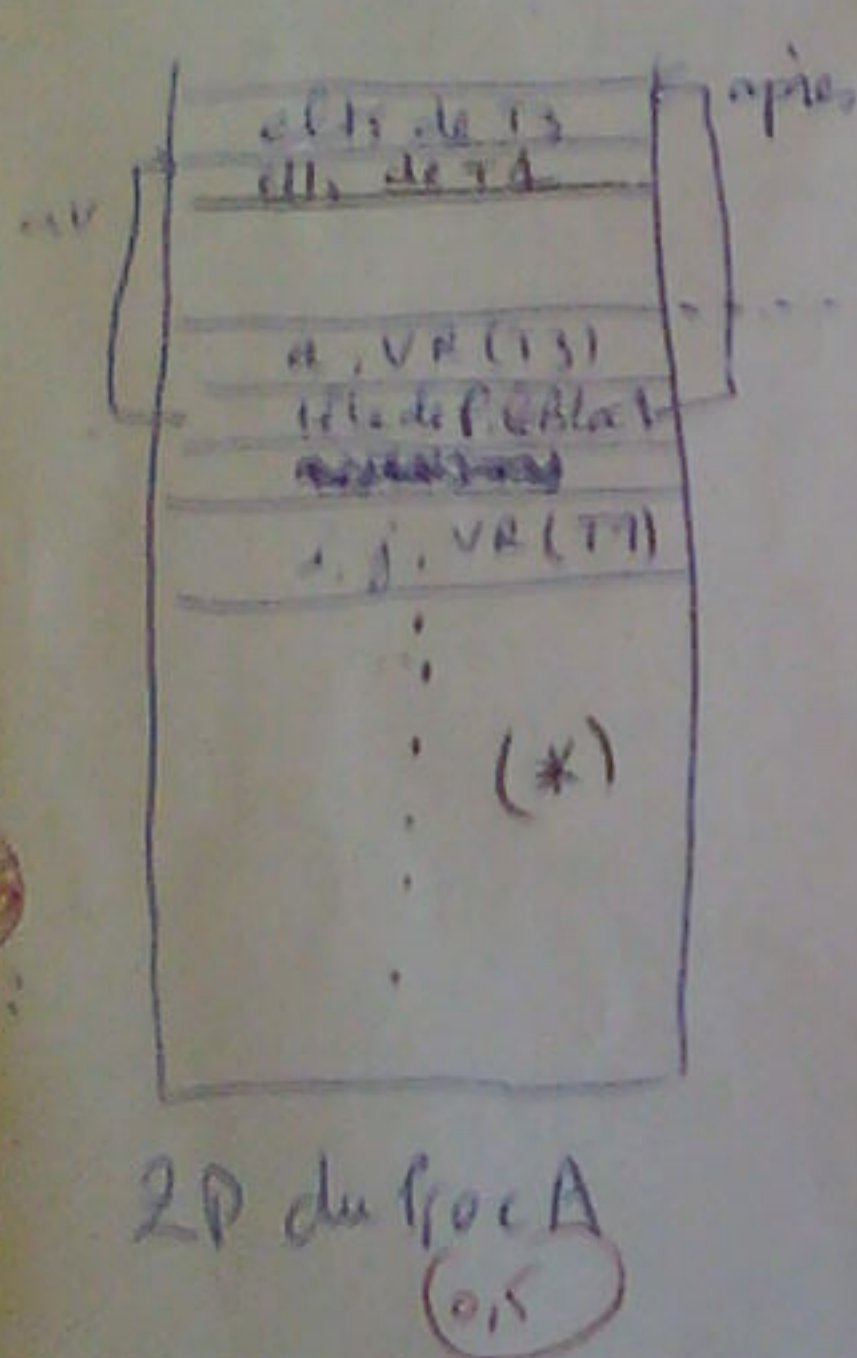
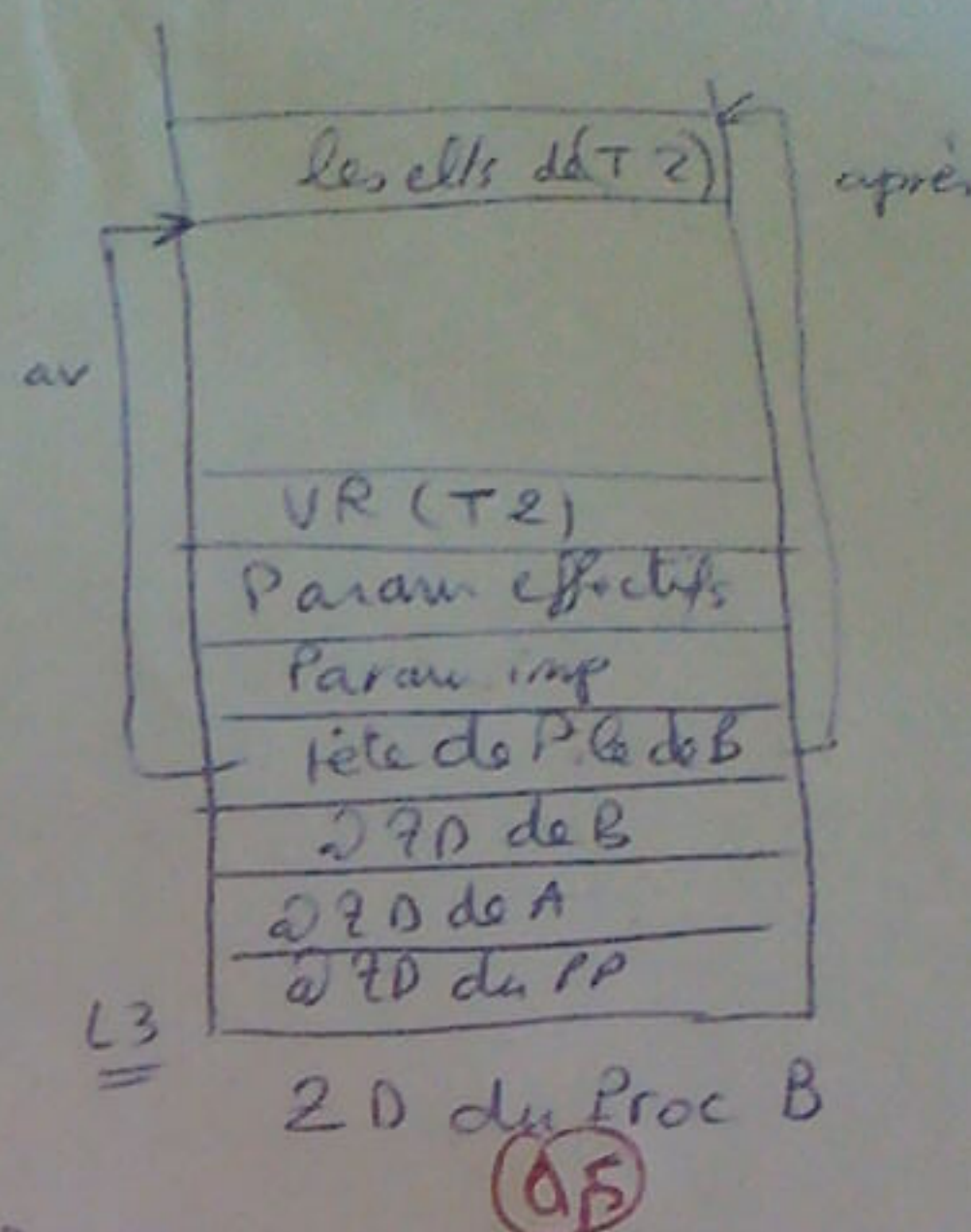
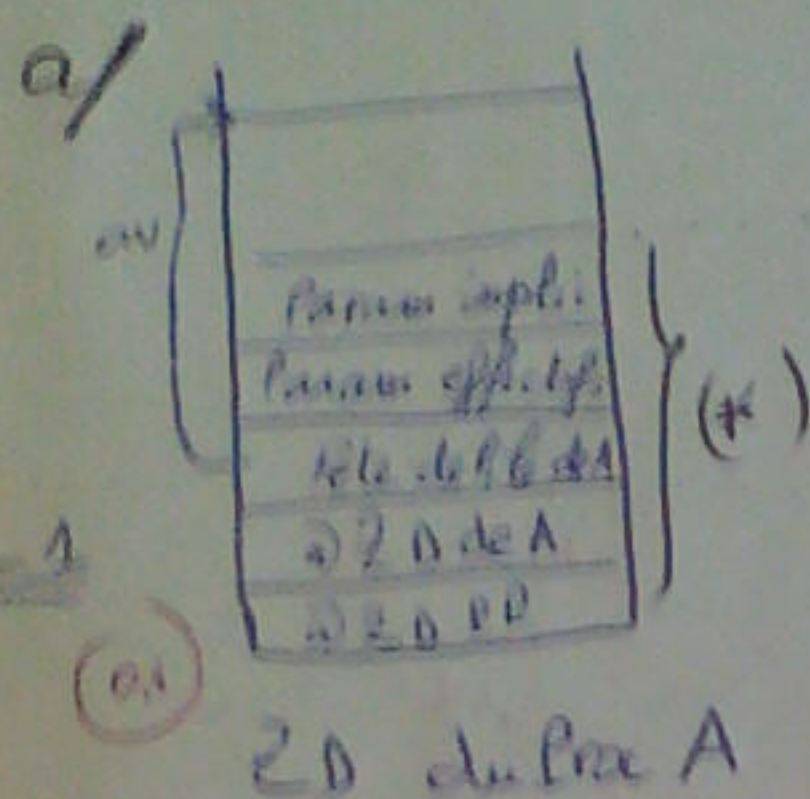
Code 6

(1pt) For $(i=0; i < n; i++)$
 $\{ a[i] := a[i] + c/y; \}$ $\xrightarrow[\text{à l'intérieur de la boucle}]{\text{déplacement du code}}$ For $(i=0; i < n; i++)$
 $\{ a[i] := a[i] + x; \}$
 2° $z := 1$
 $k := z - 1;$
 $x := y * z + k;$ $\xrightarrow[\text{code inutile}]{\text{propagation de}}$ $z := 1;$
 $k := 1 - 1;$ $\xrightarrow[\text{algébrique}]{\text{simplification}}$ $k := 0$
 $x := y * 1 + k;$ $\xrightarrow[\text{code inutile}]{\text{algébrique}}$ $x := y + k;$

$\boxed{x := y}$

$\xrightarrow[\text{+ elim code inutile}]{\text{simplif alg}}$ $k := 0$
 $x := y + 0$ $\xleftarrow[\text{de cst}]{\text{propagation}}$

Exo 3 (8 pts)



Les adresses absolues

$$T_1[K] := T_2[K, c] + x$$

- $\alpha K := c(\text{actives} + 1) + \text{deplac} + 1$ de $K / 2D$ du LP
- $\alpha c := c(\text{actives} + 1) + \text{deplac}$ de $c / 2D$ du Proc B
- $\alpha x := c(\text{actives} + 1) + 1 + \dots$ " " $x / 2D$ du Proc A.
- $\alpha T_1[K] := c(\text{actives} + 1) + \text{deplac}$ / $2D$ de Proc A : $T_1(3) + (K-3) * \text{Taille d'elt}$
- $\alpha T_2[K, c] := c(\text{actives} + 1) + \text{deplac}$ de $T_2[1, x] / 2D$ de Proc B : $\underbrace{(K-1)}_{\text{nb elts précédents}} \underbrace{(y+1)}_{\text{nbre de col}} + \underbrace{(c-x)}_{\text{deplac}}$

$$\text{car } \begin{cases} T_1(3, y) \\ T_2(1, x, x: x+y) \end{cases}$$

c/ Générer le code Objet (2pt)

$$a := x + \frac{(y * l) / k}{5} - i * j$$

Get in Acc	Acc	Code
	Vide	
Get in Acc (y, l)	y	Load y
	$T1 = y * l$	Mult l
	$T2 = T1 / k$	Div k
Get in Acc (x, T2)	$T3 = x + T2$	Add x
permutation		
Get in Acc (i, j)	T3	store T3
	i	Load i
	T4	Mult j
	T4	Store T4
Get in Acc (T3)	T3	load T3
	T4	Sub T4
		Store a

Exo 2

a/ Les relations 2pt

$A[i, j] \neq 0$ si $A[i, j] \in B1 \cup B2 \cup B3 \cup B4$

$A[i, j] \in B1 \Rightarrow 1 \leq j \leq \frac{n}{2}$ et $1 \leq i \leq j$

$A[i, j] \in B2 \Rightarrow 1 \leq j \leq \frac{n}{2}$ et $\frac{n}{2} < i \leq n-j+1$

$A[i, j] \in B3 \Rightarrow \frac{n}{2} < j \leq n$ et $n-j+1 \leq i \leq \frac{n}{2}$

$A[i, j] \in B4 \Rightarrow \frac{n}{2} < j \leq n$ et $j \leq i \leq n$

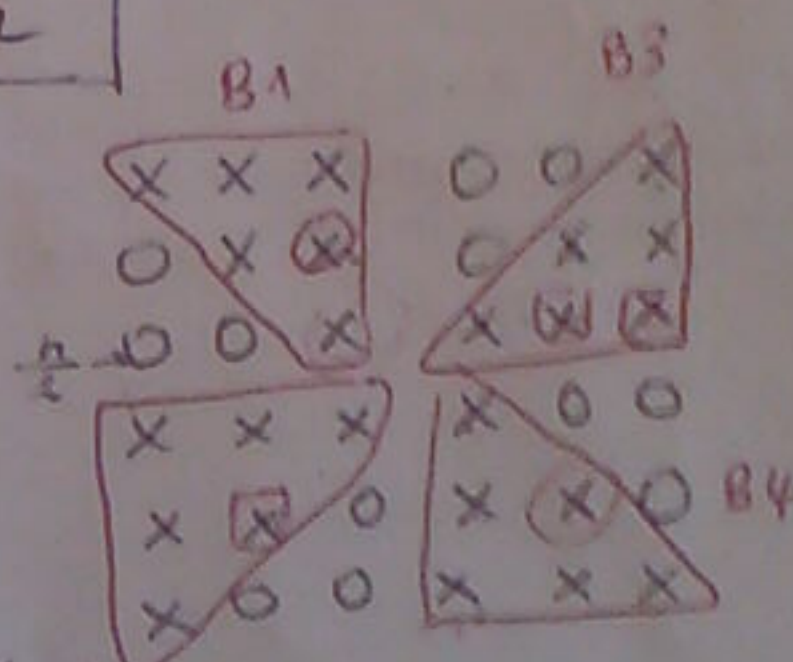
b/ Les adresses 3pt

$$a) M[i, j] \in B1 = \omega_{Base} + \left[\left(\frac{n}{2} + 1 \right) (j-1) + (i-1) \right] * \text{Taille d'elt}$$

$$b) M[i, j] \in B2 = \omega_{Base} + \left[\left(\frac{n}{2} + 1 \right) (j-1) + \underbrace{j}_{B1} + \underbrace{i - \frac{n}{2}}_{B2} \right] * \text{Taille d'elt}$$

$$c) M[i, j] \in B3 = \omega_{Base} + \left[\left(\frac{n}{2} + 1 \right) (j-1) + \underbrace{i - (n-j+1)}_{B3} \right] * \text{Taille}$$

$$d) M[i, j] \in B4 = \omega_{Base} + \left[\left(\frac{n}{2} + 1 \right) (j-1) + \underbrace{j - \frac{n}{2}}_{B3} + \underbrace{(i-j)}_{B4} \right] * \text{Taille}$$



et non connecté d'elt