

::: Architecture :::

semaine n°4
Exercices

Parcours d'un tableau :

Exemple : calculer la somme des éléments d'un tableau de 10 octets

TAB1: DB 3,2,12,1,25,40,18,8,9,4

NB: DB 10 ; nombre d'éléments

Solution n°1 :

```
int funSum(char * tab, int len)  
int i = 0;  
int sum = 0;  
while ( i < len ) {  
    sum += tab[i];  
    i++;  
}  
return sum;
```

Solution n°2 :

```
int funSum(char * tab, int len)  
char * pt = tab;  
int sum = 0;  
while ( pt < tab+len) {  
    sum+=*pt;  
    pt++;  
}  
return sum;
```

L'adresse de tab est passée dans WR0 et len est passé dans WR2

Le résultat sera calculé dans WR10

Parcours d'un tableau :

Exemple : calculer la somme des éléments d'un tableau de 10 octets

Solution n°1 :

int funSum(char * tab, int len)

```

                wr0      wr2
int i = 0;      wr4
int sum = 0;    wr10
while ( i < len ) {
    t = tab[i]; wr6

    sum += t;
    i++;
}
return sum;

```

funSum :

```

MOV      WR4,#0
MOV      WR10,#0
wh1:     CMP      WR4,WR2
         JGE     Ewh1
         PUSH   WR4
         ADD    WR4,WR0
         MOV    R6,@WR4
         POP   WR4
         MOVS  WR6,R6
         ADD   WR10,WR6
         INC   WR4
         JMP   WH1
ewh1:    RET

```

Parcours d'un tableau :

Exemple : calculer la somme des éléments d'un tableau de 10 octets

Solution n°2 :

int funSum(char * tab, int len)

```

                wr0      wr2
char * pt = tab;   wr4
int sum = 0;       wr10
while ( pt < tab+len) {

    sum+=*pt;

    pt++;
}
return sum;

```

funSum :

```

MOV    WR4,WR0
MOV    WR10, #0
WH1:   MOV    WR6,WR0
        ADD    WR6,WR2
        CMP    WR4,WR6
        JGE    EWH1
        MOV    R6,@WR4
        MOVS   WR6,R6
        ADD    WR10,WR6
        INC    WR4
        JMP    WH1
EWH1:  RET

```

Parcours d'un tableau :

Exemple : calculer la somme des éléments d'un tableau de **10 mots de 16 bits**

TAB1: DW 3,2,12,1,25,40,18,8,9,4

Solution n°1 :

```
int funSum(int * tab, int len)  
int i = 0;  
int sum = 0;  
while ( i < len ) {  
    sum += tab[i];  
    i++;  
}  
return sum;
```

Solution n°2 :

```
int funSum(int * tab, int len)  
int * pt = tab;  
int sum = 0;  
while ( pt < tab+len) {  
    sum+=*pt;  
    pt++;  
}  
return sum;
```

Parcours d'un tableau :

Exemple : calculer la somme des éléments d'un tableau de 10 mots de 16b

Solution n°1 :

int funSum(int * tab, int len)

```

                wr0      wr2
int i = 0;      wr4
int sum = 0;    wr10
while ( i < len ) {
    t = tab[i]; wr6

    sum += t;
    i++;
}
return sum;

```

funSum :

```

MOV      WR4,#0
MOV      WR10,#0
wh1:    CMP      WR4,WR2
        JGE      Ewh1
        PUSH     WR4
        ADD      WR4,WR4
        ADD      WR4,WR0
        MOV      WR6,@WR4
        POP      WR4
        ADD      WR10,WR6
        INC      WR4
        JMP      WH1
ewh1:   RET

```

Parcours d'un tableau :

Exemple : calculer la somme des éléments d'un tableau de 10 octets

Solution n°2 :

int funSum(int * tab, int len)

```

                wr0      wr2
int * pt = tab; wr4
int sum = 0;    wr10
while ( pt < tab+len) {

```

```

    sum+=*pt;
    pt++;
}
return sum;

```

funSum :

```

MOV    WR4,WR0
      MOV    WR10, #0
WH1:  MOV    WR6,WR2
      ADD    WR6,WR6
      ADD    WR6,WR0
      CMP    WR4,WR6
      JGE    EWH1
      MOV    WR6,@WR4
      ADD    WR10,WR6
      INC    WR4,#2
      JMP    WH1
EWH1: RET

```

Généralisons :

Accéder à tab[i]
pour **char** tab[];
R4 = tab[WR0]

MOV WR0,i
PUSH WR0

ADD WR0,#tab

MOV **R4**,@WR0

POP WR0

Accéder à tab[i]
pour **int** tab[];
WR4=tab[WR0]

(variante)

MOV WR0,i
PUSH WR0

ADD WR0,WR0

ADD WR0,#tab

MOV **WR4**,@WR0

POP WR0

Accéder à tab[i]
pour **long** tab[];
DR4=tab[WR0]

MOV WR0,i
PUSH WR0

ADD WR0,WR0

ADD WR0,WR0

ADD WR0,#tab

MOV **WR4**,@WR0

MOV **WR6**,@WR0+2

POP WR0

Exercice du TP3 :

Exemple : inverser un tableau

Solution n°1 :

Utiliser 2 pointeurs :

le premier part du début du tableau

le second part de la fin du tableau

Tant que les pointeurs ne se croisent pas, leur contenu est permuté et ils sont déplacés l'un vers l'autre ...

=> Nous appellerons TAB le tableau de int et N le nombre d'éléments du tableau

=> Ecrire l'algorithme en utilisant des POINTEURS

Exercice du TP3 :

Exemple : inverser un tableau

Solution n°1 : ETAPE 1 - ALGO

```
int * d = TAB
int * f = TAB + sizeof(int)*(N-1)
while ( d < f ) {
    int v0 = *d
    int v1 = *f
    *f = v0;
    *d = v1;
    d += sizeof(int);
    f -= sizeof(int);
}
```

Exercice du TP3 :

Exemple : inverser un tableau

Solution n°1 : ETAPE 2 - SIMPLIFICATION

```
int * d = TAB
int * f = N-1
    f = f* sizeof(int)
    f = f+TAB
while ( d < f ) {
    int v0 = *d
    int v1 = *f
    *f = v0;
    *d = v1;
    d += sizeof(int);
    f -= sizeof(int);
}
```

Exercice du TP3 :

Exemple : inverser un tableau

Solution n°1 : ETAPE 3 – AFFECTATION DES REGISTRES

int * d = TAB	D : WR0	WR0 = #TAB
int * f = N	F : WR2	WR2 = N
f = f - 1		WR2 = WR2 - 1
f = f * sizeof(int)		WR2 = WR2 + WR2
f = f + TAB		WR2 = WR2 + #TAB
while (d < f) {		while (WR0 < WR2) {
int v0 = *d	v0 : WR4	WR4 = @WR0
int v1 = *f	v1 : WR6	WR6 = @WR2
*f = v0;		@WR2 = WR4
*d = v1;		@WR0 = WR6
d += sizeof(int);		WR0 = WR0 + 2
f -= sizeof(int);		WR2 = WR2 - 2
}		}

Exercice du TP3 :

Exemple : inverser un tableau

Solution n°1 : ETAPE 4 – TRADUCTION FINALE

WR0 = #TAB		MOV	WR0,#TAB
WR2 = N		MOV	WR2,N
WR2 = WR2 -1		DEC	WR2
WR2 = WR2 + WR2		ADD	WR2, WR2
WR2 = WR2 + #TAB		ADD	WR2,#TAB
while (WR0 < WR2) {	WH0:	CMP	WR0,WR2
		JGE	EWH0
WR4 = @WR0		MOV	WR4,@WR0
WR6 = @WR2		MOV	WR6,@WR2
@WR2=WR4		MOV	@WR2,WR4
@WR0=WR6		MOV	@WR0,WR6
WR0 = WR0 + 2		INC	WR0,#2
WR2 = WR2 - 2		DEC	WR2,#2
}		JMP	WH0
	EWH0:	RET	

Exercice du TP3 :

Exemple : inverser un tableau

Solution n°2 :

Utiliser la pile :

La pile est une structure dans laquelle le premier élément ajouté est le dernier retiré ... elle inverse donc l'ordre...

Proposer un algorithme ...

Exercice du TP3 :

Exemple : inverser un tableau

Solution n°2 : utilisation de la pile - ALGORITHME

```
for ( i = 0 ; i < NB ; i ++ ) empiler tab[i];  
for ( i = 0 ; i < NB ; i ++ ) tab[i] = depiler;
```

Exercice du TP3 :

Exemple : inverser un tableau

Solution n°2 : utilisation de la pile - SIMPLIFICATION

```
int i = 0
while ( i < NB) {
    v = tab[i]
    empiler v
    i ++
}
i = 0
while ( i < NB) {
    v = depiler
    tab[i] = v
    i++
}
```

Exercice du TP3 :

Exemple : inverser un tableau

Solution n°2 : utilisation de la pile – AFFECTATION DES REGISTRES

int i = 0	WR0	WR0 = 0
while (i < NB) {		while (WR0 < NB) {
v = tab[i]	WR2	WR2 = TAB[WR0]
empiler v		empiler WR2
i ++		WR0++
}		}
i = 0;		WR0 = 0
while (i < NB) {		while (WR0 < NB) {
v = depiler		WR2 = dépiler
tab[i] = v		TAB[WR0] = WR2
i++		WR0 ++
}		}

Exercice du TP3 :

Exemple : inverser un tableau

Solution n°2 : utilisation de la pile – TRADUCTION

WR0 = 0		MOV	WR0,#0
while (WR0 < NB) {	WH1:	CMP	WR0,NB
		JGE	EWH1
WR2 = TAB[WR0]		PUSH	WR0
		ADD	WR0,WR0
		ADD	WR0,#TAB
		MOV	WR2,@WR0
		POP	WR0
empiler WR2		PUSH	WR2
WR0++		INC	WR0
}		JMP	WH1
WR0 = 0	EWH1:	MOV	WR0,#0
while (WR0 < NB) {	WH2:	CMP	WR0,NB
		JGE	EWH2
WR2 = dépiler		POP	WR2
TAB[WR0] = WR2		PUSH	WR0
		ADD	WR0,WR0
		ADD	WR0,#TAB
		MOV	@WR0,WR2
		POP	WR0
WR0 ++		INC	WR0
}		JMP	WH2
	EWH2:	RET	