

## TD6 : Assembleur et pipeline logiciel IA-64

### Instructions conditionnelles

Donner le code IA-64 pour le programme suivant :

```
int T[N], a=0, b=0, c=0, d=0, i;
for (i=0; i<N; i++) {
    if (T[i] <5)
        a++;
    if ((T[i] >=5)&& (T[i] <10))
        b++;
    if ((T[i] >=10)&& (T[i] <15))
        c++;
    if (T[i] >=15)
        d++; }
```

### Pipeline logiciel : exemple 1

Donner une version optimisée avec pipeline logiciel du code

```
int X[200], Y[200], Z;
for (i=0 ; i<200 ; i++)
    Y[i]=X[i]+Z;
```

### Pipeline logiciel : exemple 2

Donner une version optimisée avec pipeline logiciel sans et avec déroulage de boucle préalable pour le code

```
int X[200], Y[200], Z[200];
for (i=0 ; i<200 ; i++)
    Y[i]=X[i]+Z[i];
```

### Pipeline logiciel : exemple 3

Donner une version optimisée avec pipeline logiciel du code

```
float X[200], Y[200], Z[200];
for (i=0 ; i<200 ; i++)
    if (X[i]>=Y[i])
        Z[i]=X[i];
    else Z[i] = Y[i];
```

### Pipeline logiciel : exemple 4

Donner une version optimisée avec pipeline logiciel du code

```
int X[200], Y[200], Z, T;
Z=200
while (Z>=0){
```

```
if (X[Y[i]]==T) break;
    else Z--;
```

### **Pipeline logiciel : exemple 5**

Donner le code optimisé avec pipeline logiciel pour

```
float X[200], Y[200], Z[200];
for (i=0; i<200; i++)
    Z[i]=X[i]+Y[i];
```

### **Pipeline logiciel : exemple 6**

Donner une version optimisée avec pipeline logiciel du code

```
float X[200], Y[200], Z;
for (i=0; i<200; i++)
    Z+=X[i]*Y[i];
```

### **Pipeline logiciel : exemple 7**

Donner le code pour

```
int X[200], Y[200];
for (i=1; i<200; i++)
    Y[i]=X[i]+X[i+1];
```

### **Pipeline logiciel : exemple 8**

Donner le code optimisé avec pipeline logiciel pour

```
float X[200], Y[200], Z[200], T;
for (i=0; i<200; i++){
    Z[k+CT] = X[k] + T*Y[k];
    T= Z[k+CT] -T;}
```

### **Annexe**

Latence des instructions

Consommateur (à droite) Producteur (en bas)	UAL	Adresse load store	Store données	Inst. flottantes	getf	setf
Instructions UAL	1	1	1			1
getf			5			5
setf			6			
Instructions flottantes			4	4	4	
Load entier	N	N+1	N		N	N
Load flottant	M+1		M+1	M+1	M+1	M+1

N=1 pour cache L1D, N=5 pour L2, N=12-15 pour L3, N=180-220 pour MP  
M = 5 pour L2, M=12-15 pour L3, M=180-220 pour MP