

III. Les procédures :

1. Définition :

.....
.....
.....

2. Déclaration d'une procédure :

En analyse :

```
DEF PROC nom_procedure (p.f1 :type1 ;..... ;p.fn :typen)
```

```
Résultat=
```

```
.....  
.....
```

```
Fin nom_procedure
```

En algorithme :

```
0) DEF Proc nom_procedure (p.f1 :type1 ;..... ;p.fn :typen)
```

```
1) .....
```

```
2) .....
```

```
.
```

```
.
```

```
n) Fin nom_procedure
```

En pascal :

```
Procedure nom_procedure (p.f1 :type1 ;... ;p.fn :typen);
```

```
Var
```

```
Begin
```

```
.
```

```
.
```

```
.
```

```
End ;
```

3. Appel d'une procédure :

En analyse et en algorithme :

.....

En pascal :

.....

Remarque :

Les paramètres **effectifs** et les paramètres **formels** doivent s'accorder du point de vue **nombre** et **ordre**.

Leurs types doivent être identiques ou compatibles, selon le mode de passage des paramètres

Passage par valeur et par variable :

Passage par valeur

Le transfert d'informations par valeur est effectué dans un seul sens, du programme principal vers le sous programme (PP → SP). Il permet au programme appelant de transmettre une valeur au programme appelé

P.appelant (PP) → P.appelé (SP)

Passage par variable

Le transfert d'informations par variable est effectué dans les 2 sens (PP ↔ SP). Toute **modification** du paramètre formel variable **entraîne automatiquement la modification** de la valeur du paramètre effectif, puisque les 2 paramètres (formel et effectif) désignent une **même variable**. Il faut précéder le paramètre formel par le mot

P.appelant → P.appelé
 ←