

U.E. Informatique 1

Travaux pratiques

Introduction au Turbo-Pascal



Gilles TALADOIRE

Février 2005

Plan

Mise en route du Turbo Pascal pour Windows	1
Utilisation de Turbo Pascal	1
Utilisation des fenêtres	2
Utilisation de l'éditeur	3
Processeurs et codification	8
Le langage Pascal	11
Structure d'un programme	11
Codage des actions	12
L'affectation	12
L'action conditionnelle	12
Les actions itératives	13
Types prédéfinis	14
Entrées et sorties élémentaires	14
Compléments	16
Compilation et exécution d'un programme	17
Procédures	18
Déclaration d'une procédure (Définition) :	18
Appel d'une procédure (Utilisation) :	19
Fonctions	21
Déclaration d'une fonction (Définition) :	21
Appel d'une fonction (Utilisation) :	22
Quelques fonctions prédéfinies	24
Tableaux	25
Déclaration d'une variable de type tableau	25
Utilisation d'une variable de type tableau	25
Déclaration de TYPES	26
Le schéma itératif POUR	27
Notion de type en PASCAL	28
Le schéma CAS ... DE	29
Fichiers	30

Mise en route du Turbo Pascal pour Windows

Les fichiers du logiciel Turbo Pascal se trouvent dans le répertoire "C:\Program Files\TPW". Le fichier exécutable est dans le répertoire "C:\Program Files\TPW" et se nomme TPW.EXE.

donc pour lancer ce programme, plusieurs méthodes :

- en utilisant les menus : **Démarrer / Programmes / Turbo Pascal**
- en utilisant le raccourci sur le bureau : **Turbo Pascal**

Un **menu** horizontal s'affiche en haut de l'écran avec des **options** ainsi que des icônes utilisées comme "raccourcis".

Utilisation de Turbo Pascal

Présentation de l'écran (cf. schéma page suivante)

Le **menu principal** et les **icônes** "raccourcis" en haut de la fenêtre.

Le **bureau** et ses **fenêtres**,

La **barre d'état** (en bas de la fenêtre) qui indique la ligne et la colonne du curseur, l'état du fichier (modifié ou pas), le mode de frappe (Insertion/Suppression) ainsi que différents messages (erreurs, ..).

Pour choisir une option dans un menu horizontal ou vertical :

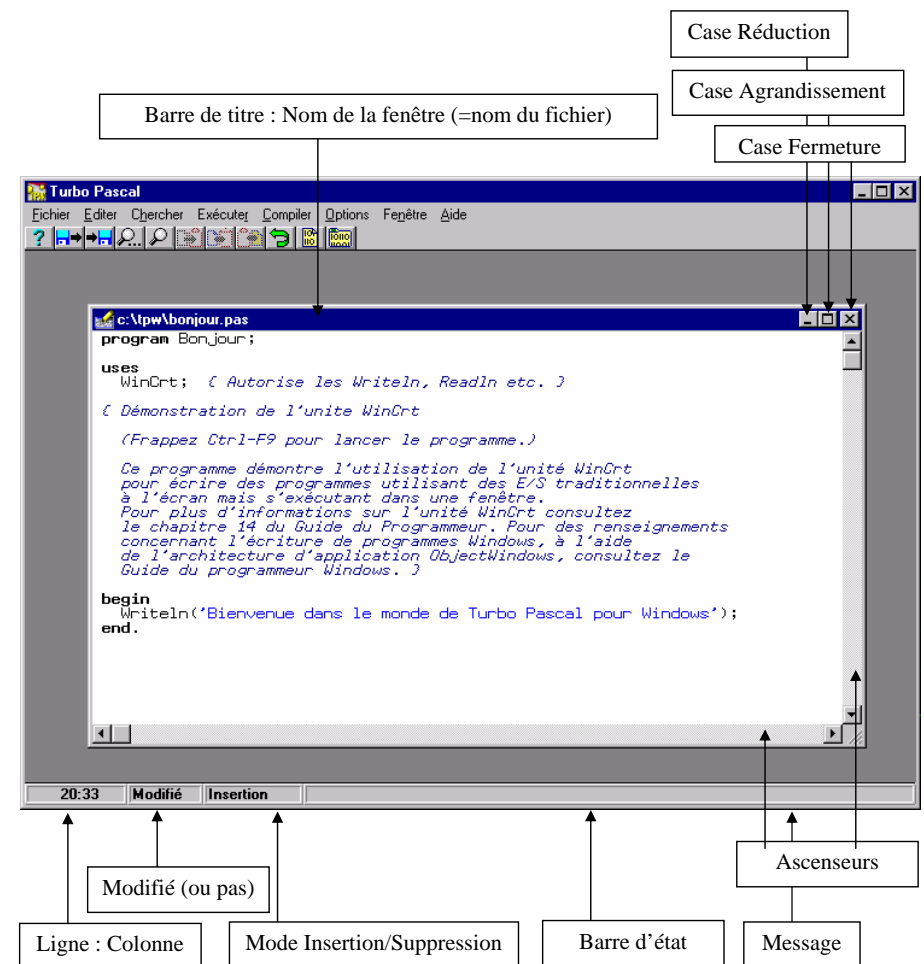
- Méthode 1 :
 - appuyer sur "ALT" (ou "F10"),
 - se positionner sur l'option dans le menu horizontal à l'aide des flèches →, ←, dans le menu vertical à l'aide des flèches ↑, ↓
 - valider en appuyant sur "Entrée".
- Méthode 2 : Appuyer sur "ALT" et la lettre soulignée de l'option ou directement sur les touches raccourcis signalées dans les menus
- Méthode 3 : Utiliser la souris pour dérouler les menus ou directement pour cliquer sur l'icône "raccourci"

A tout moment, on peut avoir de l'**aide** en appuyant sur la touche "F1" et "CTRL-F1".

Pour annuler l'opération en cours, appuyer sur la touche "Echap".

Pour revenir au menu principal, taper sur la touche "F10" ou bien sur la touche "ALT"

Utilisation des fenêtres



Opérations sur les fenêtres (comme sous Windows):

Déplacement : en cliquant sur la barre de titre et en déplaçant la souris (avec le clavier ALT - / Déplacement + les flèches)

Changement de la taille : en cliquant sur les bordures et en déplaçant la souris

(avec le clavier ALT - / Dimension + les flèches)

Utilisation de l'éditeur

Editeur = traitement de texte limité mais spécialisé à la saisie des programmes

Choisir l'option "**Fichier / Nouveau**", une fenêtre est créée. Nous pouvons maintenant travailler avec l'éditeur de texte, dans la fenêtre "**noname00.pas**".

Il nous est possible à présent de taper du texte comme avec une machine à écrire avec majuscules, minuscules, chiffres, caractères de ponctuation, ...

Mais l'avantage de l'éditeur (et des traitements de texte en général) par rapport à une machine à écrire, c'est que vous pouvez modifier ce texte après l'avoir tapé, pour cela vous pouvez vous déplacer avec les flèches, effacer des caractères, en écrire d'autres, tout cela est assez naturel.

Le "tiret" qui clignote est appelé le **curseur** et nous indique "où on se trouve".

Voici les différentes touches ayant un rôle spécial :

- les flèches de direction (haut, bas, droite, gauche) pour se déplacer,
- "**Suppr**" (ou "**Annul**" ou "**Del**") permet d'effacer le caractère sur lequel se trouve le curseur,
- "**BackSpace**" (←) permet d'effacer le caractère à la gauche du curseur,
- "**Entrée**" (ou "**Return**") permet d'insérer une nouvelle ligne,
- "**PgUp**" et "**PgDn**" permettent d'aller respectivement d'une page vers le haut ou vers le bas,
- "**Début**" ("**Home**") et "**Fin**" ("**End**") permettent d'aller respectivement au début ou à la fin de la ligne,
- "**Ctrl-Début**" et "**Ctrl-Fin**" permettent respectivement d'aller au début et à la fin du texte.

Exercice :

Tapez quelques définitions ci-contre.

Quelques définitions

(source <http://www-rocq.inria.fr/~deschamp/www/CMTI/glossaire.html>)

Algorithmique, n. f. :

Etude de la résolution de problèmes par la mise en œuvre de suites d'opérations élémentaires selon un processus défini aboutissant à une solution.

Ardoise électronique, n. f.

Ordinateur portatif sans clavier dans lequel les données sont entrées par saisie directe sur l'écran.

Base de données, n. f.

Ensemble de données organisé en vue de son utilisation par des programmes correspondant à des applications distinctes et de manière à faciliter l'évolution indépendante des données et des programmes.

Bit, n. m.

Information représentée par un symbole à deux valeurs généralement notées 0 et 1, associées aux deux états d'un dispositif.

Bloc, n. m. (projet d'arrêté)

Groupe de données enregistré ou transmis globalement indépendamment de son contenu.

Bogue, n. f.

Défaut de conception ou de réalisation se manifestant par des anomalies de fonctionnement.

Boule de commande, n. f.

Boule qui, par rotation dans son logement, permet de déplacer sur l'écran d'une vis un curseur et d'en relever à volonté les coordonnées.

Bureautique, n. f.

Ensemble des techniques et des moyens tendant à automatiser les activités de bureau et principalement le traitement et la communication de la parole, de l'écrit et de l'image.

Bus, n. m.

Dispositif non bouclé destiné à assurer simultanément les transferts d'information entre différents sous-ensembles d'un système informatique selon des spécifications physiques et logiques communes.

Souris, n. f. (projet d'arrêté)

Dispositif de commande tenu à la main, connecté à un ordinateur, et dont le déplacement sur une surface entraîne le déplacement d'un repère sur l'écran.

Logiciel, n. m.

Ensemble des programmes, procédés et règles, et éventuellement de la documentation, relatifs au fonctionnement d'un ensemble de traitement de données.

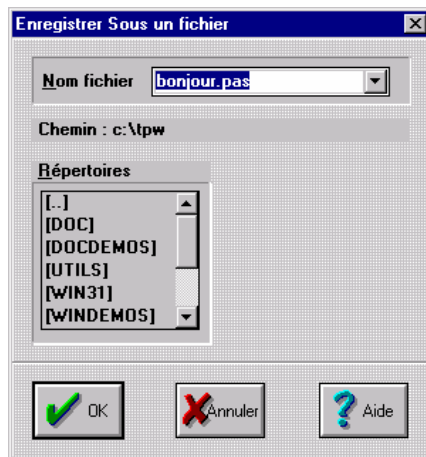
Interface, n. f.

Le texte est maintenant tapé, il est en mémoire centrale, si il y a une panne de courant, tout est à recommencer, pour éviter cela il faut le sauvegarder, le sauver, l'enregistrer, le ranger sur la disquette de travail.

Pour enregistrer un document

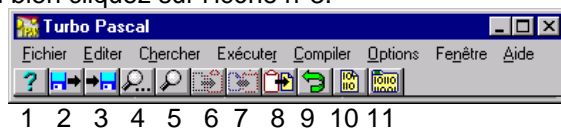
★ **La première fois**, il faut donner un nom au fichier qui va contenir le texte avec l'option "**Fichier / Enregistrer sous...**".

Une fenêtre va apparaître à l'écran vous demandant l'unité, le répertoire et le nom à donner au fichier, donnez lui ce nom (8 caractères max. + extension). Si l'extension n'est pas précisée, le système ajoutera automatiquement l'extension ".PAS" lorsque vous taperez sur la touche "Entrée".



★ **Ensuite**, quand votre fichier a un nom (celui ci s'inscrit en haut à droite de la fenêtre de travail : xxxxxxxx.PAS), il suffit d'appeler la commande "**Fichier / Enregistrer**".

Ou bien cliquez sur l'icône n°3.



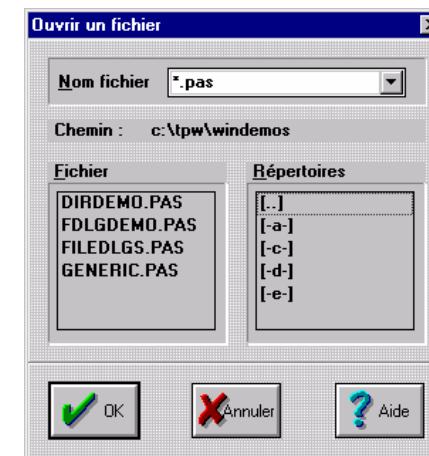
De la même manière, on va pouvoir lire, charger, ouvrir un fichier précédemment sauvegardé.

Pour ouvrir un document existant

Utilisez l'option "**Fichier / Ouvrir...**" ou l'icône n°2.

Une fenêtre va apparaître à l'écran vous demandant le nom du fichier à ouvrir :

- soit vous le lui donnez et vous tapez "**Entrée**",
- soit vous aller choisir votre fichier dans la liste proposée :
 - avec le clavier : "**Tab**" puis les flèches puis "**Entrée**",
 - avec la souris : cliquez 2 fois sur le nom du fichier.



Pour marquer un bloc

(Pour sélectionner une partie d'un document)

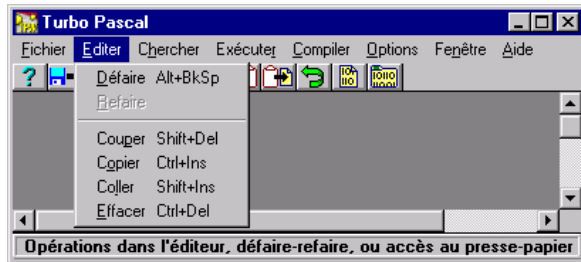
En utilisant le clavier :

- placez le **curseur au début de la zone** (sur le premier caractère),
- tapez "**Ctrl K - B**",
- amener le **curseur à la fin de la zone** à l'aide des touches de direction,
- tapez "**Ctrl K - K**", le bloc marqué est en inverse vidéo

En utilisant **la souris** :

- placez le **curseur au début de la zone** (sur le premier caractère),
- cliquez et restez cliqué,
- amener le **curseur à la fin de la zone** à l'aide des touches de direction,
- relâchez la souris, le bloc marqué est en inverse vidéo

Utilisation du presse-papiers



"**Editer / Couper**" : Copie la zone sélectionnée dans le presse-papiers puis la supprime du texte

"**Editer / Copier**" : Copie la zone sélectionnée dans le presse-papiers

"**Editer / Coller**" : Ajoute le contenu du presse-papiers à l'emplacement du curseur

"**Editer / Effacer**" : Supprime la zone sélectionnée

"**Défaire**" : Annule la dernière opération

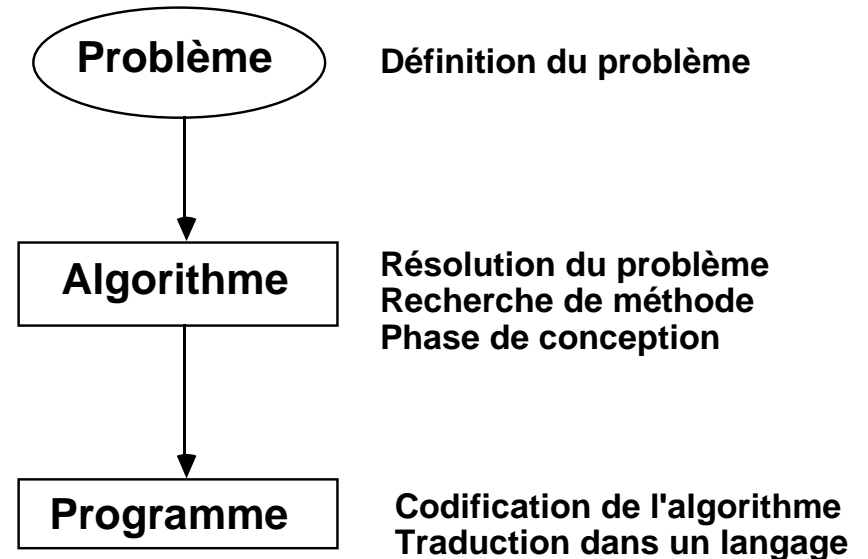
"**Refaire**" : Reproduit la dernière opération

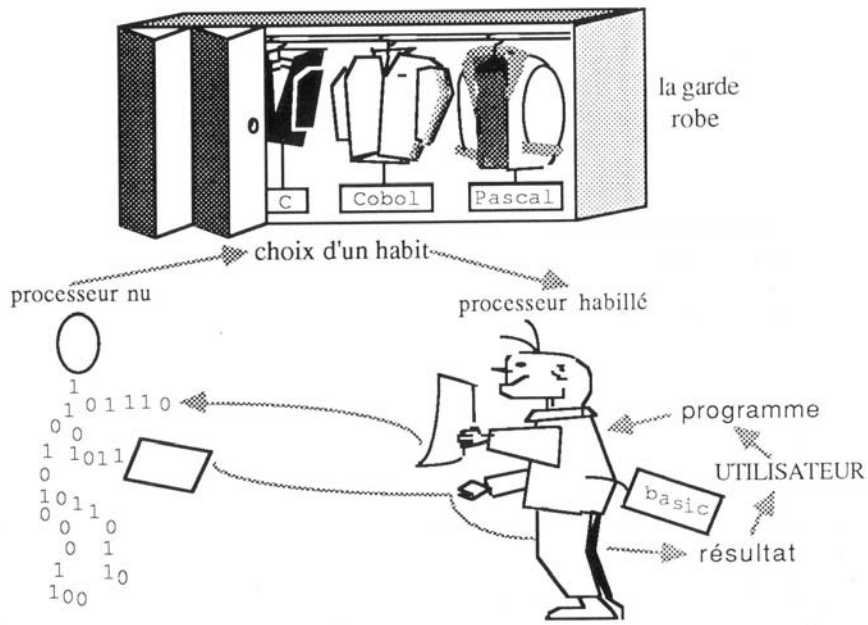
Quand vous avez fini de travailler, il faut sortir du logiciel.

Pour quitter TURBO

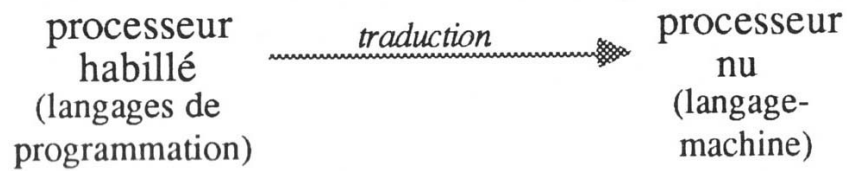
- utilisez l'option "**Fichier / Quitter**" ou "**ALT - F4**"

Processeurs et codification

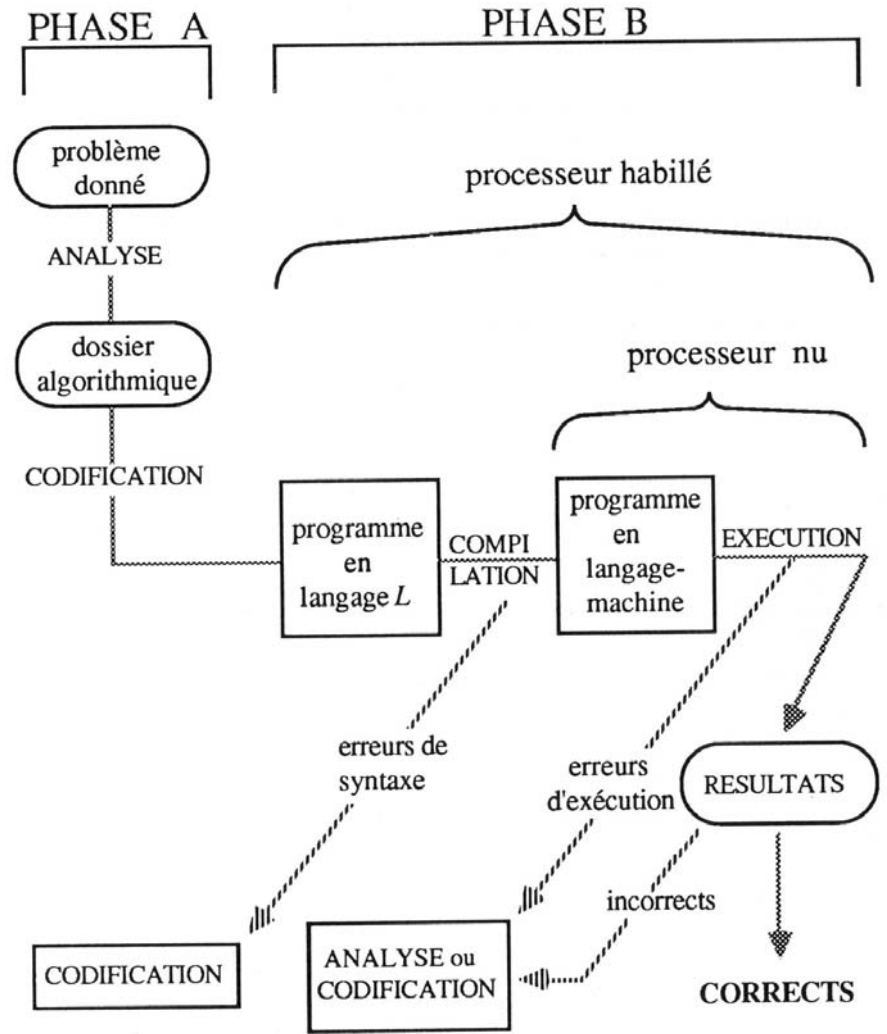




Processeur nu et processeur habillé



Passage du processeur habillé au processeur nu



Etape de la résolution d'un problème

Le langage Pascal

Structure d'un programme

- 3 parties :
- Déclaration des données, (Constantes, Types et Variables)
 - Déclaration des procédures,
 - Corps du programme.

```
PROGRAM nomduprogramme ;  
(* Deux méthodes pour mettre des commentaires : *)  
{ Déclaration des données }  
CONST identificateur = constante ;  
    ...  
TYPE    nomtype = descripteurde type ;  
    ...  
VAR     nomvar <, nomvar > ... : type ;  
    ...  
{ Déclaration des procédures }  
...  
{ Corps du programme }  
BEGIN  
...  
END.
```

Codage des actions

Le corps du programme est une succession d'énoncés séparés par des points-virgules ";".

L'affectation

nomvar ← expression
devient

```
nomvar := expression
```

L'action conditionnelle

Si Condition
Alors Action1
Sinon Action2
Finsi
devient

```
IF condition  
THEN BEGIN  
    Enoncé1  
END  
ELSE BEGIN  
    Enoncé2  
END
```

Attention, si les actions 1 et 2 sont élémentaires (1 seule action), le parenthésage avec les mots-clés BEGIN et END peut être évité :

```
IF condition  
THEN Enoncé1  
ELSE Enoncé2
```

Les actions itératives

Tantque

Tantque condition faire
Action
Finfaire

devient

```
WHILE condition DO  
BEGIN  
  Énoncé  
END
```

Attention, si l'action est élémentaire (1 seule action), le parenthésage avec les mots-clés BEGIN et END peut être évité :

```
WHILE condition DO  
  Énoncé
```

Répéter

Répéter
Action
Jusque condition
devient

```
REPEAT  
  Énoncé  
UNTIL condition
```

Remarque : L'énoncé peut être simple ou composé.

Types prédéfinis

INTEGER

valeur entière (-32768 ... 32767)

REAL

valeur réelle

BOOLEAN

valeur logique (TRUE et FALSE)

CHAR

caractère

STRING[n]

chaîne de caractères

Entrées et sorties élémentaires

lire nomvar

devient

```
READLN(nomvar)
```

afficher nomvar 'texte ...'

devient

```
WRITE(nomvar, 'texte ...')
```

Pour afficher un saut de ligne (pour passer à la ligne suivante), on utilisera la procédure WRITELN avec ou sans arguments.

Exemple

Division entière

de DVDE (dividende) par DVSR (diviseur)

Résultat dans Q (quotient) et R (reste)

```
PROGRAM DIVISION;

USES WINCRT ;

VAR DVDE, DVSR, Q, R : INTEGER;

BEGIN
  Write('Donner le dividende : '); ReadLn(DVDE);
  Write('Donner le diviseur : '); ReadLn(DVSR);
  R := DVDE;
  Q := 0;
  WHILE R >= DVSR DO
  BEGIN
    R := R - DVSR;
    Q := Q + 1;
  END;
  Write('Le résultat de la division entière de ', DVDE, '
par ', DVSR, ' est ', Q, ', il reste ', R, '.');
END.
```

Compléments

• Opérateurs de comparaison

Opérateurs vus en cours	Opérateurs en Pascal
=	=
≠	<>
<	<
≤	<=
>	>
≥	>=

- Les opérateurs arithmétiques sont les mêmes que ceux vus en cours.
- Pour l'utilisation de Windows, ajouter " USES WINCRT " au début du programme.
- Fonction Rac(x) : réel → Fonction SQRT(X) : Real (Square Root = racine carrée)
- Fonction SQR(x) : Real (Square = x²)
- Apostrophe : dans une chaîne, il faut deux " quotes " (= apostrophe) pour faire un apostrophe
Exemple : writeln('l'année 2000')
- Affichage de réels
Write(X : a : b) → X va être affiché sur a caractères au minimum avec b décimales.

Compilation et exécution d'un programme

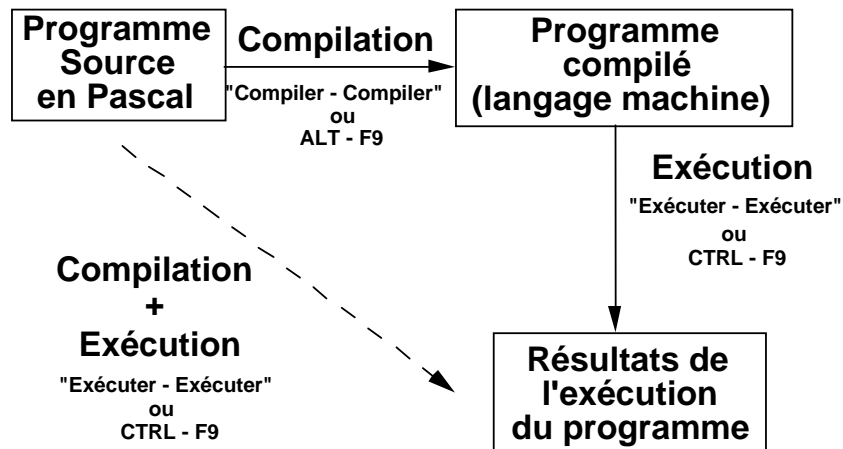
Une fois que votre programme est saisi, il faut le **compiler**.

Pour cela, choisir l'option "**Compiler / Compiler**" (ou "**ALT - F9**")

Ensuite, il faut **exécuter** le programme résultant de la compilation.

Pour cela, choisir l'option "**Exécuter / Exécuter**" (ou "**CTRL - F9**").

Par la suite, si vous demandez directement l'exécution du programme, celui-ci sera d'abord compilé puis exécuté, vous pourrez donc éviter la première étape (invisible mais toujours présente).



Procédures

Déclaration d'une procédure (Définition):

Dans la zone "**Déclaration des procédures**" (voir p. 11), après les déclarations des variables et avant le programme principal

Sous la forme :

```
PROCEDURE nomproc(liste de paramètres formels);  
  
{Déclaration des variables locales}  
VAR    v1 : T1;  
      ...  
  
{ Corps de la procédure }  
BEGIN  
...  
END;
```

où la liste des paramètres formels se présente sous la forme :

```
P1 : T1; P2, P3 : T2;  
      pour les paramètres données
```

```
VAR P4 : T4; VAR P5 : T5
```

```
      pour les paramètres résultats  
      et les paramètres données/résultats
```

Procédures

Appel d'une procédure (Utilisation) :

Dans le programme principal ou dans le corps d'une autre procédure, on écrira :

```
Begin
  ...
  nomproc(liste des paramètres effectifs);
  ...
End;
```

où la liste des paramètres effectifs est une suite de constantes, de variables et d'expressions correspondants aux paramètres séparés par des virgules ","

Un appel à une procédure est considéré comme une action élémentaire.

Exemple

```
Program DIVISION1;
{ Déclaration des variables globales
  (du programme principal) }
VAR   A, B, RESTE1, RESTE2 : INTEGER;

{ Déclaration des procédures }
Procedure CALCULRESTE(DVDE, DVSR : INTEGER;
  VAR R : INTEGER);

{ Déclaration des variables locales à la procédure }
VAR   Q : INTEGER;

{ Corps de la procédure }
Begin
  Q := DVDE div DVSR;
  R := DVDE - (Q * DVSR);
End;

{ Programme principal }
Begin
  Write('Donnez le dividende : '); Readln(A);
  Write('Donnez le diviseur : '); Readln(B);

  { Appels à la procédure CalculReste }
  CALCULRESTE(A, B, RESTE1);
  CALCULRESTE(10, 3, RESTE2);

  Writeln('Le reste de la division entière de ', A,
' par ', B, ' est ', RESTE1);
  Writeln('Le reste de la division entière de 10 par ', 3
est ', RESTE2);
End.
```

Fonctions

Déclaration d'une fonction (Définition):

Dans la zone "Déclaration des procédures" (voir p. 11), après les déclarations des variables et avant le programme principal

Sous la forme :

```
FUNCTION nomfonc(Paramètres formels) : Type;  
  
{Déclaration des variables locales}  
VAR   v1 : T1;  
      ...  
  
{ Corps de la fonction }  
BEGIN  
  ...  
END;
```

où la liste des paramètres formels se présente sous la même forme que pour une procédure,

où Type est le type du résultat que fournit la fonction.

Fonctions

Appel d'une fonction (Utilisation) :

Dans le programme principal ou dans le corps d'une autre procédure, on écrira :

```
Begin  
  ...  
  X := nomfonc(liste des paramètres effectifs);  
  ...  
End;
```

où la liste des paramètres effectifs est une suite de constantes, de variables et d'expressions correspondants aux paramètres séparés par des virgules ","

Un appel à une fonction est considéré comme une expression.

Exemple

```
Program DIVISION2;
{ Déclaration des variables globales
  (du programme principal) }
VAR   A, B, RESTE1 : INTEGER;

{ Déclaration des procédures et fonctions }
Function RESTE(DVDE,DVSR:INTEGER) : INTEGER;

{ Déclaration des variables locales à la fonction }
VAR   Q : INTEGER;

{ Corps de la fonction }
Begin
  Q := DVDE div DVSR;
  {Passage de la valeur rendue par la fonction}
  RESTE := DVDE - (Q * DVSR);
End;

{ Programme principal }
Begin
  Write('Donnez le dividende : '); ReadLn(A);
  Write('Donnez le diviseur : '); ReadLn(B);

  { Appel à la fonction Reste }
  RESTE1 := RESTE(A, B);

  Writeln('Le reste de la division entière de ', A,
' par ', B, ' est ', RESTE1);

  { Appel à la fonction Reste }
  Writeln('Le reste de la division entière de 10 par ', '3
est ', RESTE(10, 3));
End.
```

Quelques fonctions prédéfinies

La liste des procédures et fonctions prédéfinies peut être obtenue dans l'aide (menu "Aide / Procédures et Fonctions").

Vous connaissez déjà les procédures

Read
ReadLn
Write
Writeln

un peu particulières parce qu'elles ont un nombre de paramètres variable.

Procedure Randomize;
{ Initialise le générateur de nombres aléatoires }

Function Random(range : Integer) : Integer;
{ Renvoie un nombre aléatoire }

Function Sqrt(x : Real) : Real;
{ Renvoie la racine carré de l'argument }

Function Sqr(x : Type) : même type que l'argument;
{ Renvoie le carré de l'argument }

Tableaux

Déclaration d'une variable de type tableau

```
VAR T : ARRAY [MIN..MAX] OF TYPELEMENT;
```

où MIN et MAX sont respectivement l'indice inférieur et supérieur du tableau

Remarque :

Dans la plupart des cas, MIN est égal à 1.

Exemple :

```
VAR T : ARRAY [1..10] OF INTEGER;
```

Utilisation d'une variable de type tableau

L'accès au ième élément du tableau T s'effectue de la manière suivante :

T[i]

Déclaration de TYPES

Pour éviter la répétition assez lourde de ces déclarations, nous allons pouvoir créer de nouveaux types.

La déclaration de nouveaux types s'effectue avant les déclarations de variables de la manière suivante :

```
TYPE NomType = DescriptionType;
```

Exemple :

Pour créer un type correspondant à un tableau de 10 entiers, on écrira :

```
TYPE TABINT = ARRAY [1..10] OF INTEGER;
```

Le type ainsi créé pourra ensuite être utilisé comme un nom de type standard en utilisant le nom qu'on lui a attribué.

Exemple :

La déclaration d'une variable du type TABINT s'effectuera de la manière suivante :

```
VAR T : TABINT;
```

Le schéma itératif POUR

Pour I de Id à If incrément inc Faire
<action ou suite d'actions>
Finfaire

En Pascal, ce schéma devient :

```
FOR I:=Id TO If DO  
  Énoncé
```

si l'incrément inc est égal à 1

```
FOR I:=Id DOWNTO If DO  
  Énoncé
```

si l'incrément inc est égal à -1

Il n'existe pas en Pascal de "Pour" avec d'autres incréments. Pour gérer ceux-ci, il faut utiliser le schéma "Tantque"

Remarque :

Comme pour les autres instructions Pascal, si l'énoncé n'est pas élémentaire, il faut parenthéser ces énoncés par les mots-clés BEGIN et END.

Notion de type en PASCAL

Types Intervalles

Type NomType = Vmin .. Vmax;

Types Enumérés

Type NomType = (Val1, Val2, Val3, ..., Valn);

Types Composés (ou Articles)

```
NomVar : Record  
  Champ1 : Type1;  
  Champ2 : Type2;  
  Champ3 : Type3;  
  ...  
  ChampN : TypeN;  
End;
```

La référence à un des champs de la variable s'effectuera par son nom préfixé d'un point "." et du nom de la variable :

NomVar.ChampI

On peut définir le type correspondant pour des raisons évidentes de simplification :

```
Type NomType = Record  
  ...  
End;
```

Le schéma CAS ... DE

```
CASE Expression OF
  v1 : A1;
  v2 : A2;
  v3 : A3;
  ...
  vn : AN;
END
```

En turbo-Pascal seulement :

```
CASE Expression OF
  v1 : A1;
  v2 : A2;
  v3 : A3;
  ...
  vn : AN;
else
  AE
END
```

Fichiers

Type fichier

Type NomType = **File of** TypeElem ;

Procédures fournies

Procedure Assign(f : File; nom : String);

Assigne le nom d'un fichier disque à une variable fichier.

Correspondance logique <-----> physique
programme <-----> DOS

Procedure Rewrite(Var f : File);

<====> Créer(F)

Procedure Reset(Var f : File);

Ouverture du fichier f en lecture ou écriture

Positionnement sur le premier enregistrement

<-----> Prendre(F)

Procedure Close(Var f : File);

<====> Fermer(F)

Procedure Write(f : File, V : TypeElem);

<====> EcrireArticle(F, V)

Procedure Read(f : File, V : TypeElem);

<====> LireArticle(F, V)

Function Eof(f : File) : Boolean;

<====> Fonction Fin(F)

Exercice : Ecrire une procédure en Pascal ayant l'effet de la procédure "Compléter(F)".