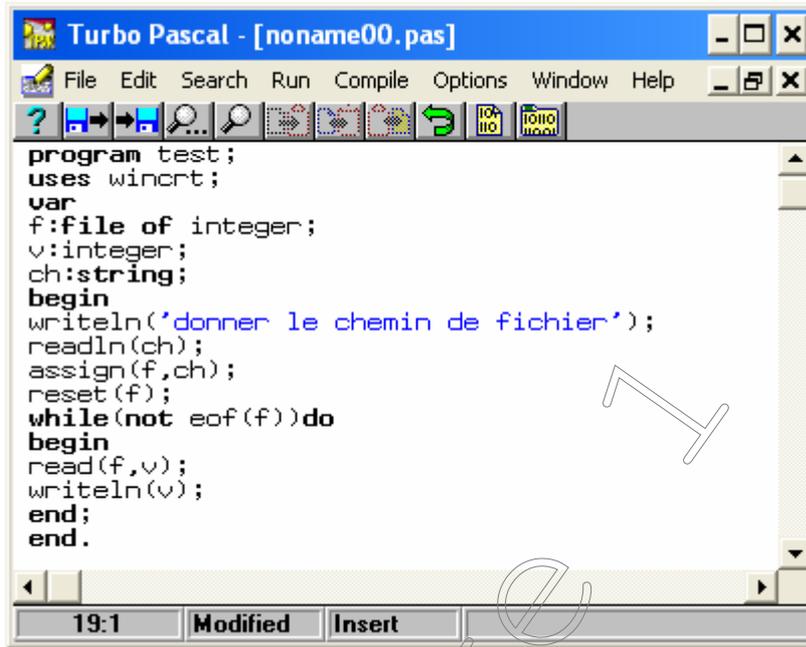


Gestion des erreurs d'entrées/sorties relatives aux fichiers

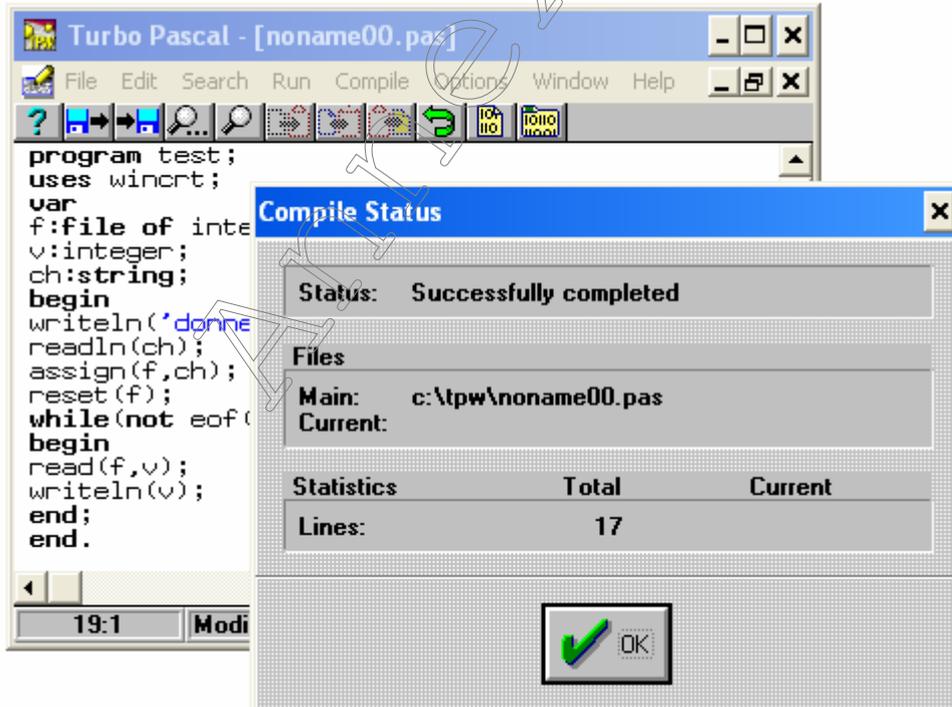
Soit le fichier **suite.dat** qui se trouve dans la partition C: , sachant que le fichier contient des valeurs, on veut écrire un programme qui affiche son contenu.

Notre programme peut se présenter comme suit :



```
program test;
uses wincrt;
var
f:file of integer;
v:integer;
ch:string;
begin
writeln('donner le chemin de fichier');
readln(ch);
assign(f,ch);
reset(f);
while(not eof(f))do
begin
read(f,v);
writeln(v);
end;
end.
```

Maintenant on va compiler le programme pour vérifier s'il y'a des erreurs et les corrigées.

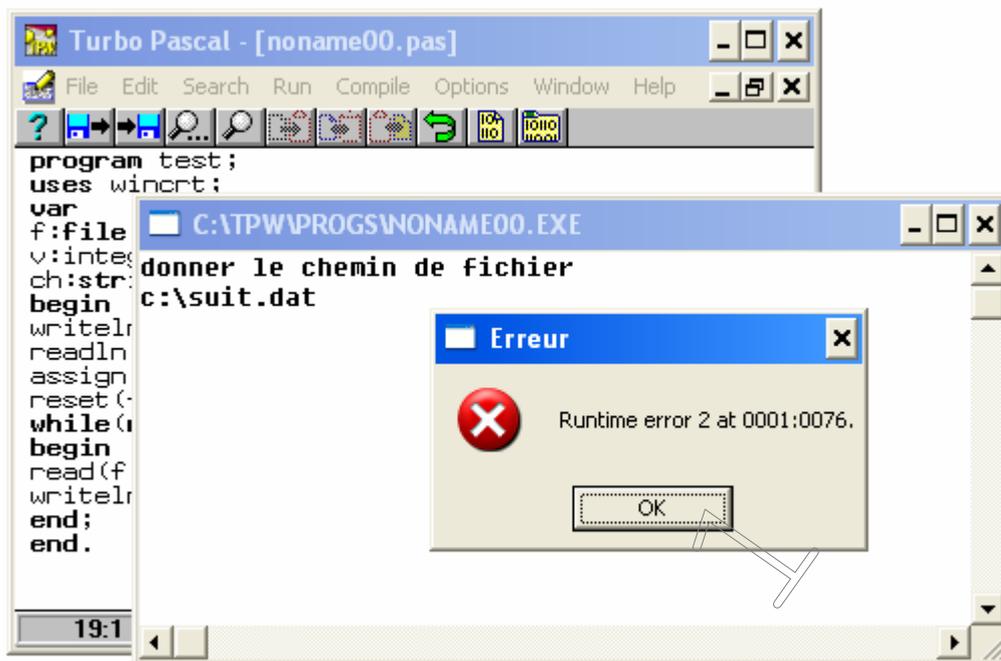


The screenshot shows the Turbo Pascal IDE with the same Pascal program as above. The 'Compile' button has been pressed, and the 'Compile Status' dialog box is open. The dialog box displays the following information:

Statistics	Total	Current
Lines:	17	

The status is 'Successfully completed'. The main file is 'c:\tpw\noname00.pas'. The dialog box also features a green checkmark and an 'OK' button.

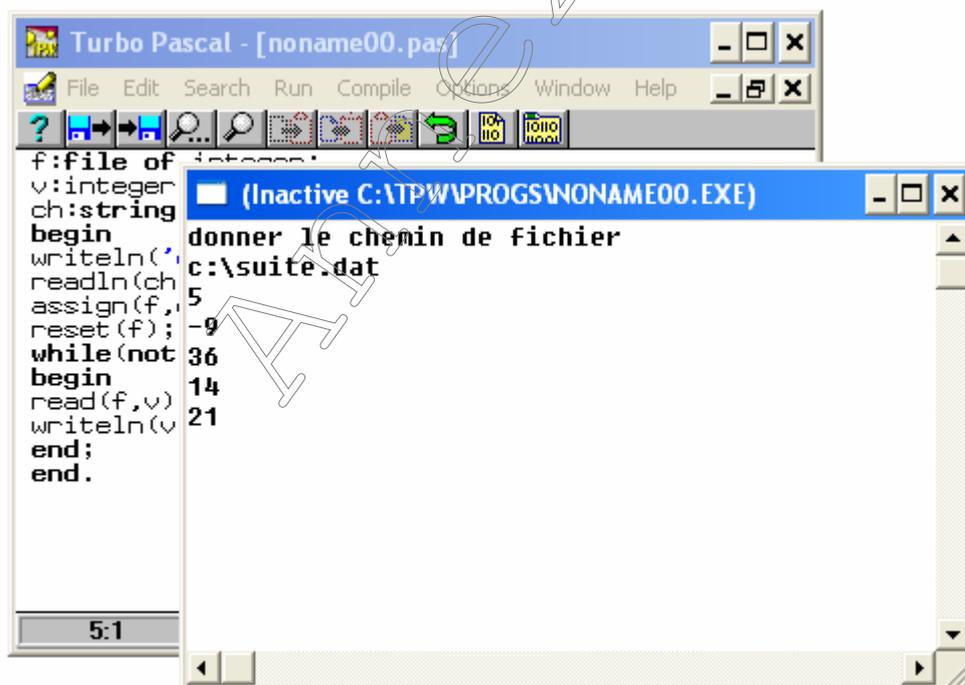
Pas des erreurs lors de la compilation, on passe donc à l'exécution de programme.



Oups ! Une erreur lors de l'exécution de programme, le programme s'arrête et n'affiche rien. Alors d'où vient cette erreur ?

Oui, c'est exacte le chemin de fichier saisi est c:\suit.dat et non pas c:\suite.dat

On va réexécuter le programme et on fait attention à ne pas saisir un chemin incorrect.



Voila maintenant tout est bien passé.

Revenons une autre fois sur l'erreur générer par le compilateur lors de l'exécution et essayons de l'étudier.

D'où vient l'erreur lors de l'exécution ?

Quand on donne un chemin de fichier qui n'existe pas (une faute de frappe par exemple), le programme, lors de la tentative d'ouverture de fichier, ne sait pas ouvrir un fichier qui n'existe pas (car **reset** ne permet pas comme **rewrite** de créer le fichier).

Est il possible de résoudre ce problème ?

Oui, c'est possible, en effet, il s'agit d'une erreur d'entrée/sortie générée par le compilateur qui en même temps nous offre la possibilité de gérer manuellement (dans le programme) et momentanément les erreurs d'entrées/sorties en utilisant la directive de compilation **\$I** et la fonction **IORESULT**.

Principe

1. Désactiver la vérification effectuée par le compilateur en ajoutant la ligne de code **{\$I-}**
2. Récupérer le numéro de l'erreur par la fonction **IORESULT** et agir selon l'erreur
3. Rendre la main au compilateur en ajoutant la ligne de code **{\$I+}**

Quelques numéros des erreurs

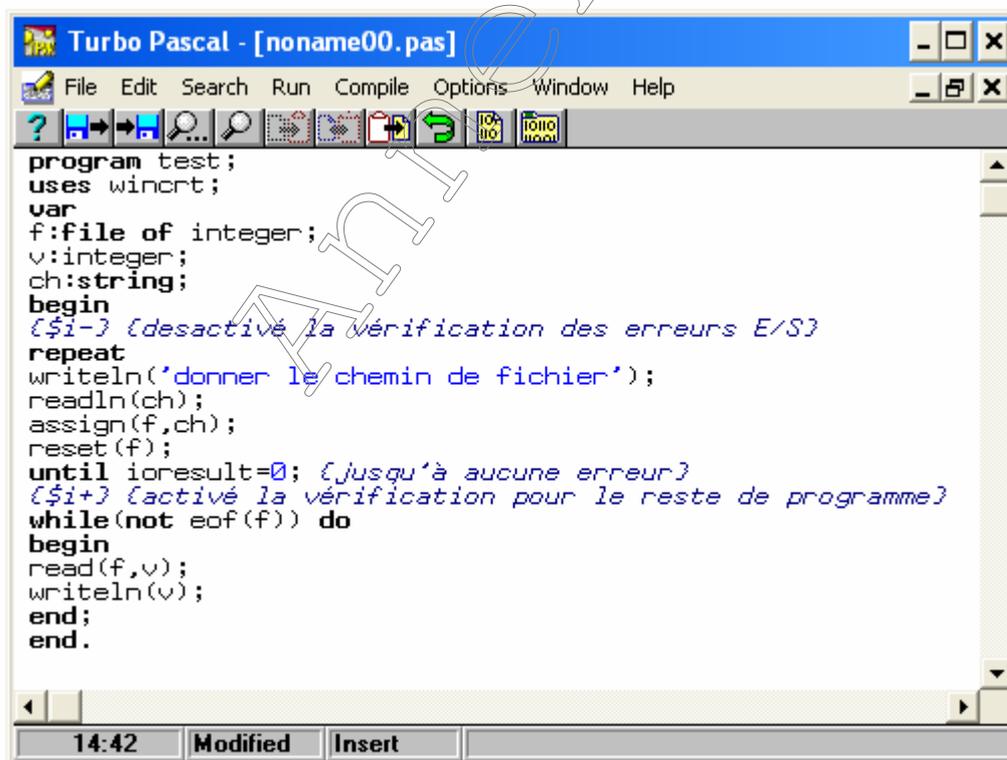
- 0 : aucune erreur
- 2 : fichier non trouvé
- 3 : chemin non valide
- 4 : plusieurs fichiers sont ouverts

Nous allons maintenant modifier notre programme pour qu'il ne s'arrête pas lorsqu'on saisit un chemin invalide.

Le programme au lieu de s'arrêter suite à une faute de frappe, va demander à l'utilisateur de saisir le chemin de nouveau.

→ Répéter la lecture de chemin jusqu'à avoir un chemin valide, donc un chemin qui ne provoque pas une erreur ceci signifie que **IORESULT** retourne la valeur 0.

Le programme doit se présenter comme suit :



```
program test;
uses wincrt;
var
  f:file of integer;
  v:integer;
  ch:string;
begin
  {$I-} {désactivé la vérification des erreurs E/S}
  repeat
    writeln('donner le chemin de fichier');
    readln(ch);
    assign(f,ch);
    reset(f);
  until ioresult=0; {jusqu'à aucune erreur}
  {$I+} {activé la vérification pour le reste de programme}
  while(not eof(f)) do
  begin
    read(f,v);
    writeln(v);
  end;
end.
```

14:42 Modified Insert

Test de programme

```
program test;
uses wincrt;
var
  f:file of integer;
  v:integer;
  ch:string;
begin
  {$i-} {desactivé la
repeat
  writeln('donner le c
  readln(ch);
  assign(f,ch);
  reset(f);
until ioresult=0; {/
  {$i+} {activé la vér
while(not eof(f)) do
begin
  read(f,v);
  writeln(v);
end;
end.
```

```
(Inactive C:\TPW\PROGS\NONAME00.EXE)
donner le chemin de fichier
n'importe quoi
donner le chemin de fichier
une faute de frappe
donner le chemin de fichier
c:\suite.dat
5
-9
36
14
21
```

14:42 Modified Insert

Dans d'autres cas, on est amené à créer le fichier au lieu de répéter la lecture de chemin, ce qui se traduit par :

```
program test;
uses wincrt;
var
  f:file of integer;
  v:integer;
  ch:string;
begin
  {$i-} {desactivé la vérification des erreurs E/S}
  writeln('donner le chemin de fichier');
  readln(ch);
  assign(f,ch);
  reset(f);
  if(ioresult<>0) then rewrite(f); {si erreur créer le fichier}
  {$i+} {activé la vérification pour le reste de programme}
  ...
  ...
end.
```