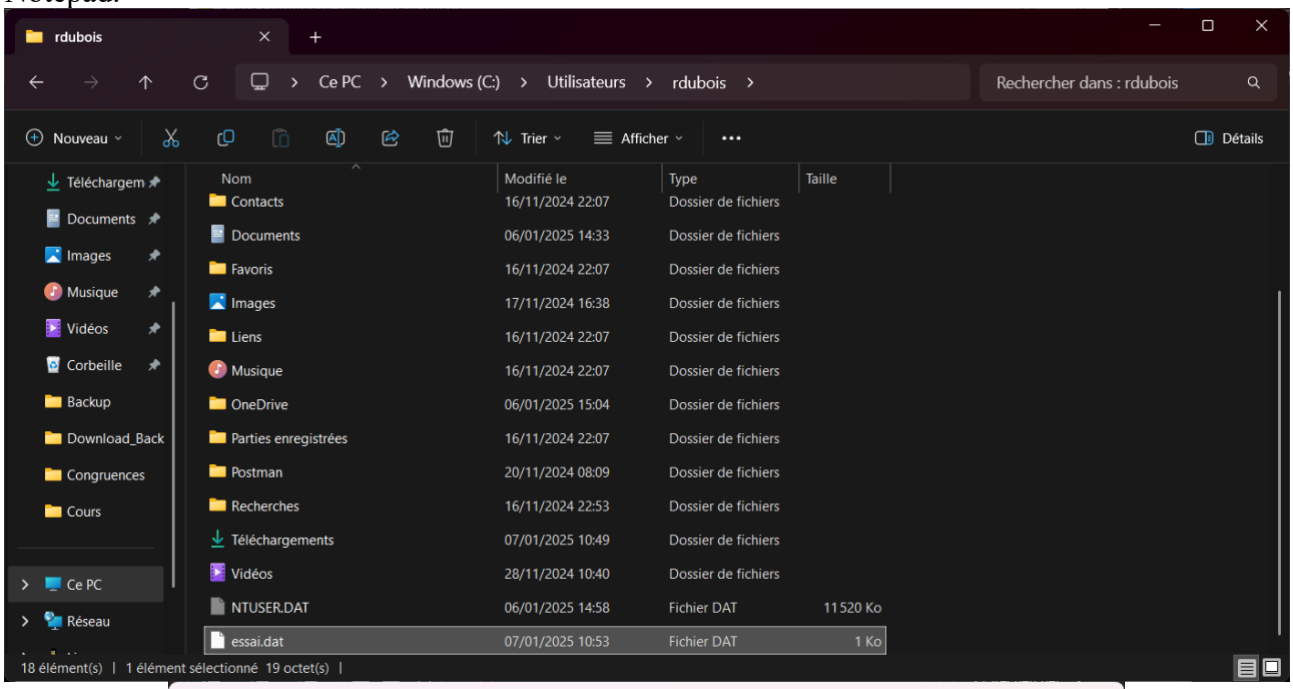
TP 3 Commandes DOS

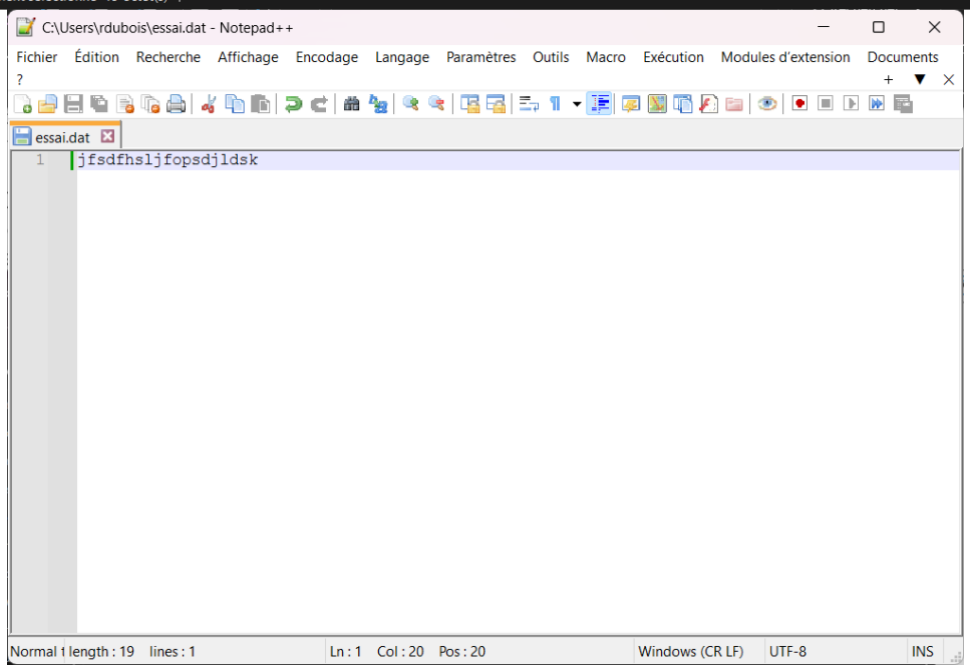
CONTEXTE

Le dépannage d'un ordinateur Windows nécessite de connaître les commandes Dos. Ce petit TP va vous faire manipuler les commandes les plus connues. Pour obtenir de l'aide sur une commande "commande /?". Rappel des commandes précédentes avec les flèches. Complétion automatique avec les tabulations. Commande *cls* (clear screen) pour effacer l'écran. Commande *exit* pour sortir.

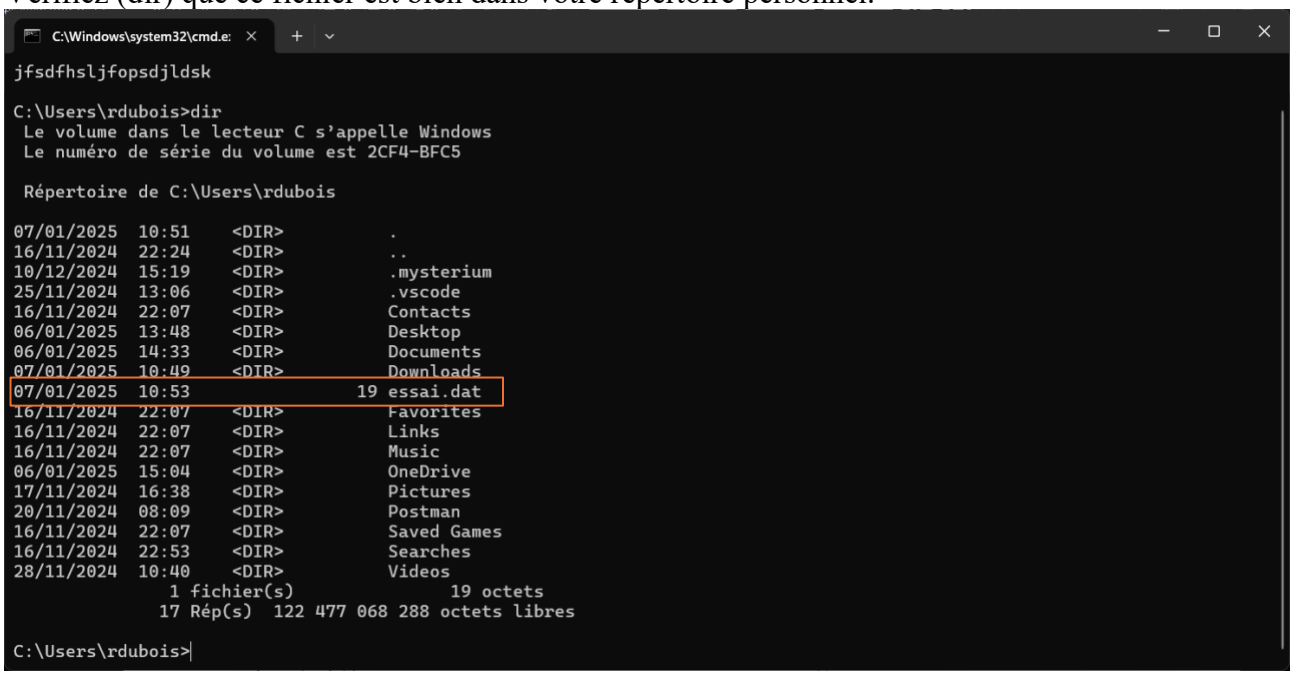
GESTION DES FICHIERS

Avec le gestionnaire de fichiers, créez un fichier essai.dat et mettez dedans du texte avec le Notepad.

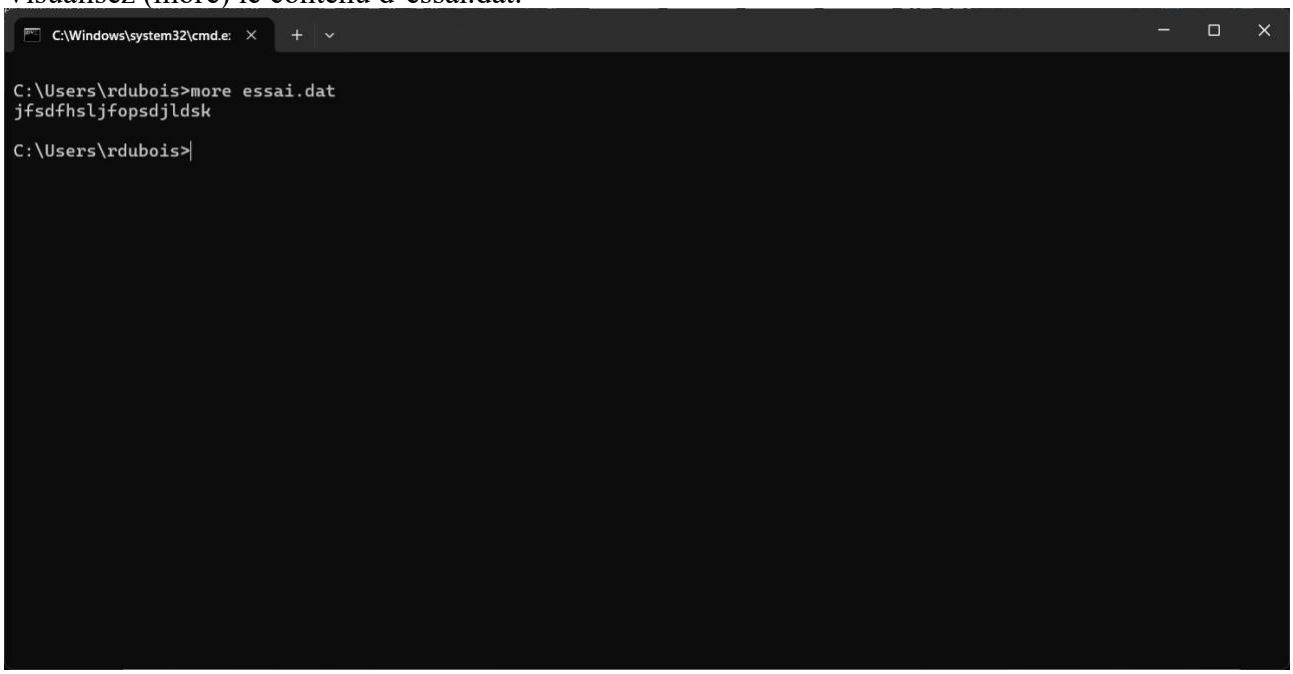




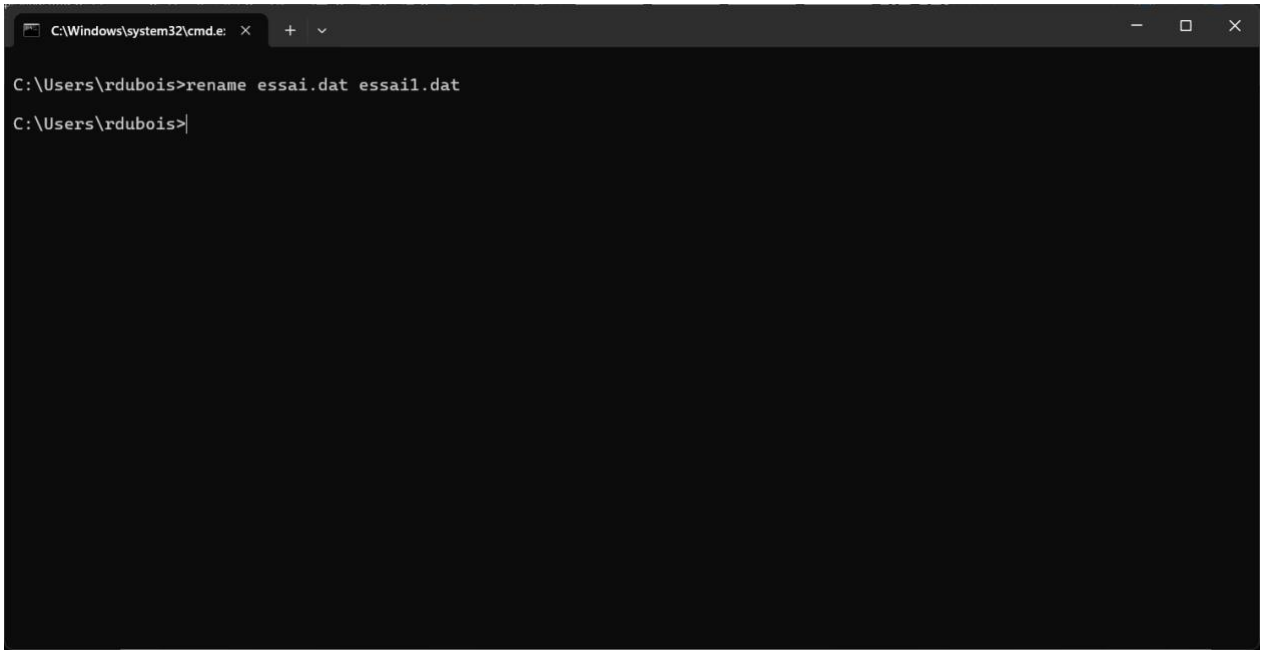
Vérifiez (dir) que ce fichier est bien dans votre répertoire personnel.



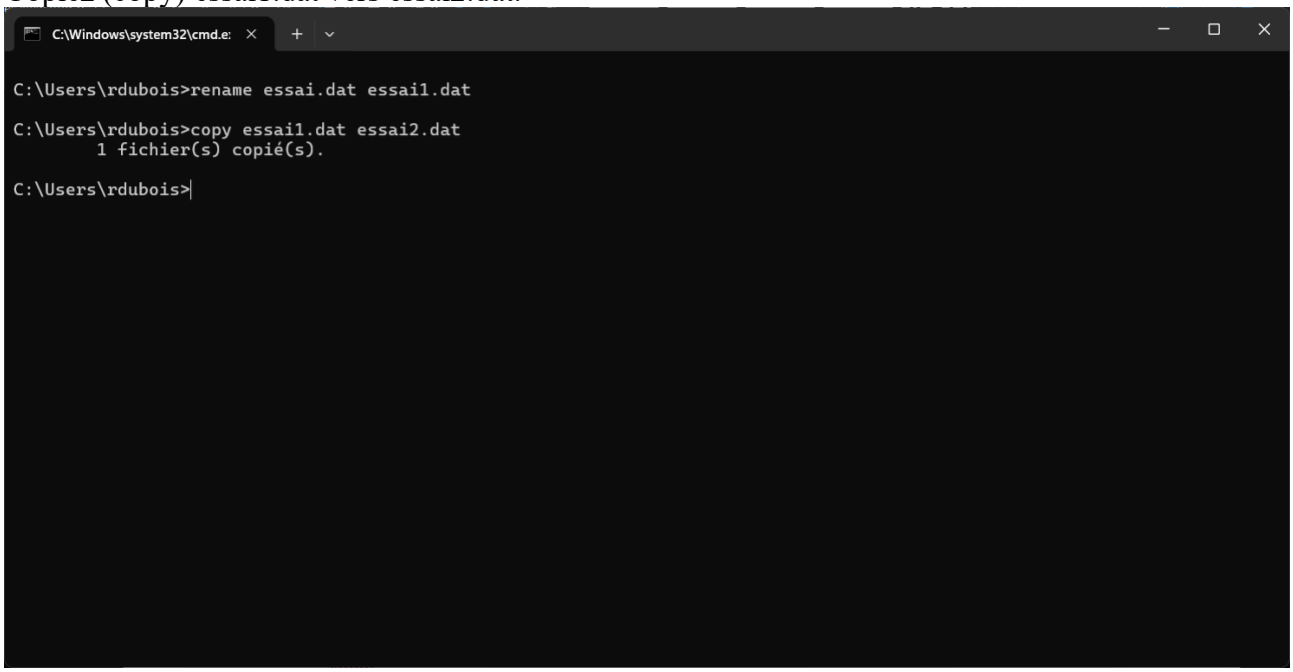
Visualisez (more) le contenu d'essai.dat.



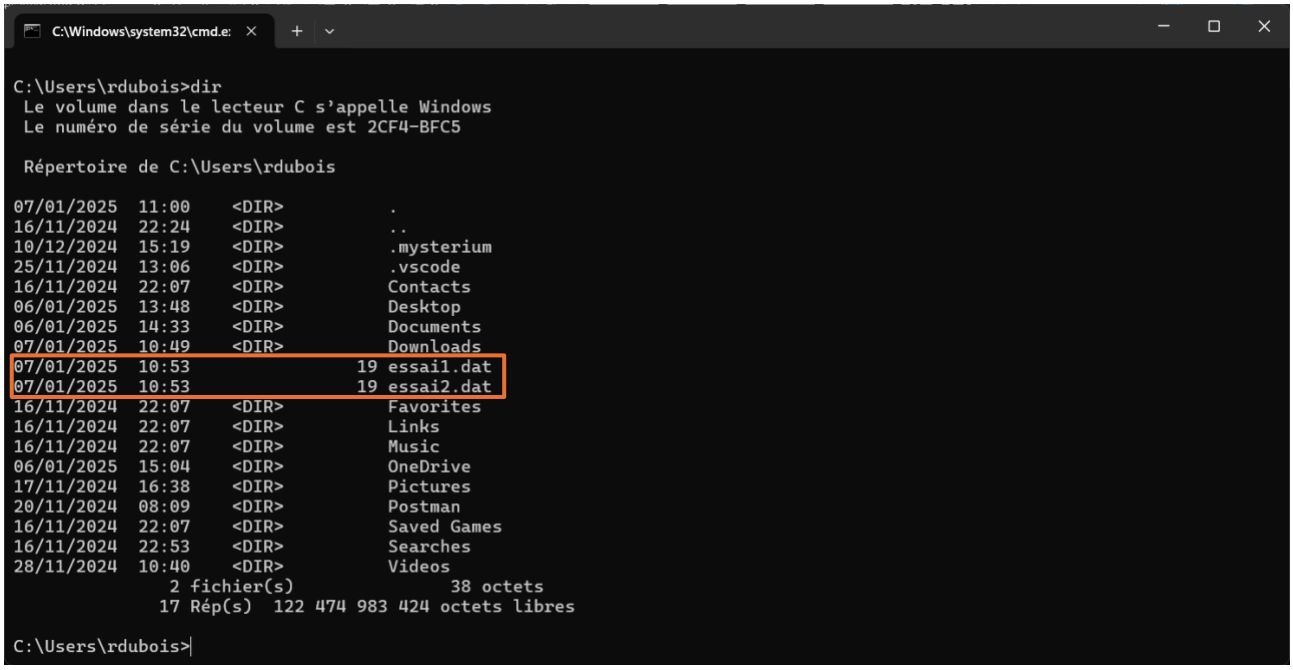
Changez le nom (rename) d'essai.dat en essai1.dat.



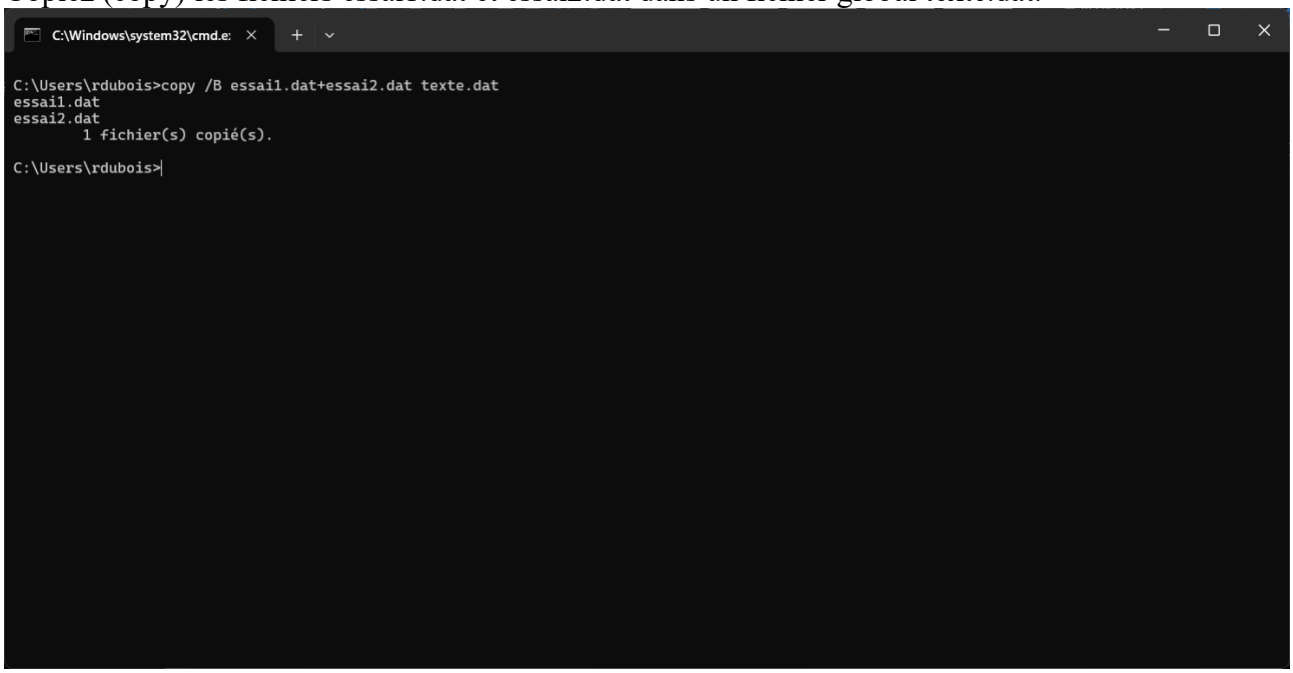
Copiez (copy) essai1.dat vers essai2.dat.



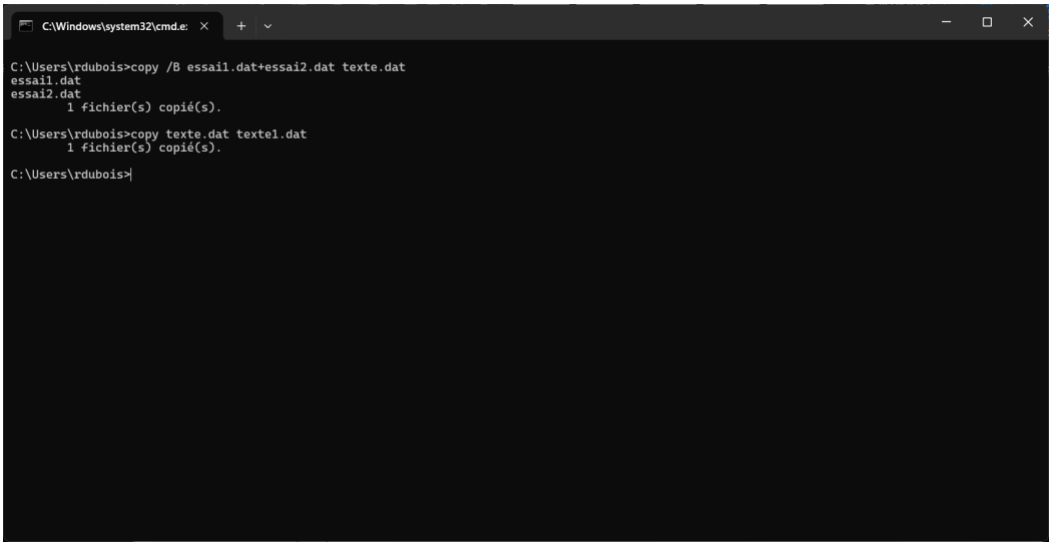
Visualisez (dir) tous les fichiers « .dat » de votre répertoire et uniquement ceux-ci.



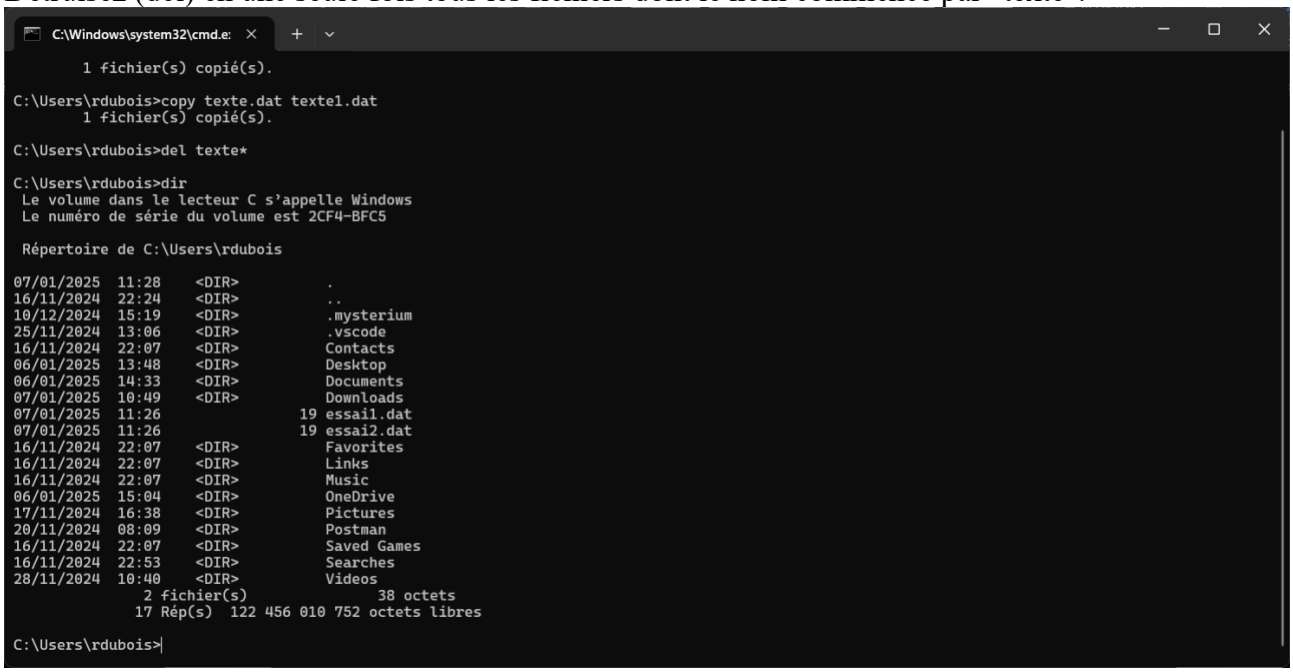
Copiez (copy) les fichiers essai1.dat et essai2.dat dans un fichier global texte.dat.



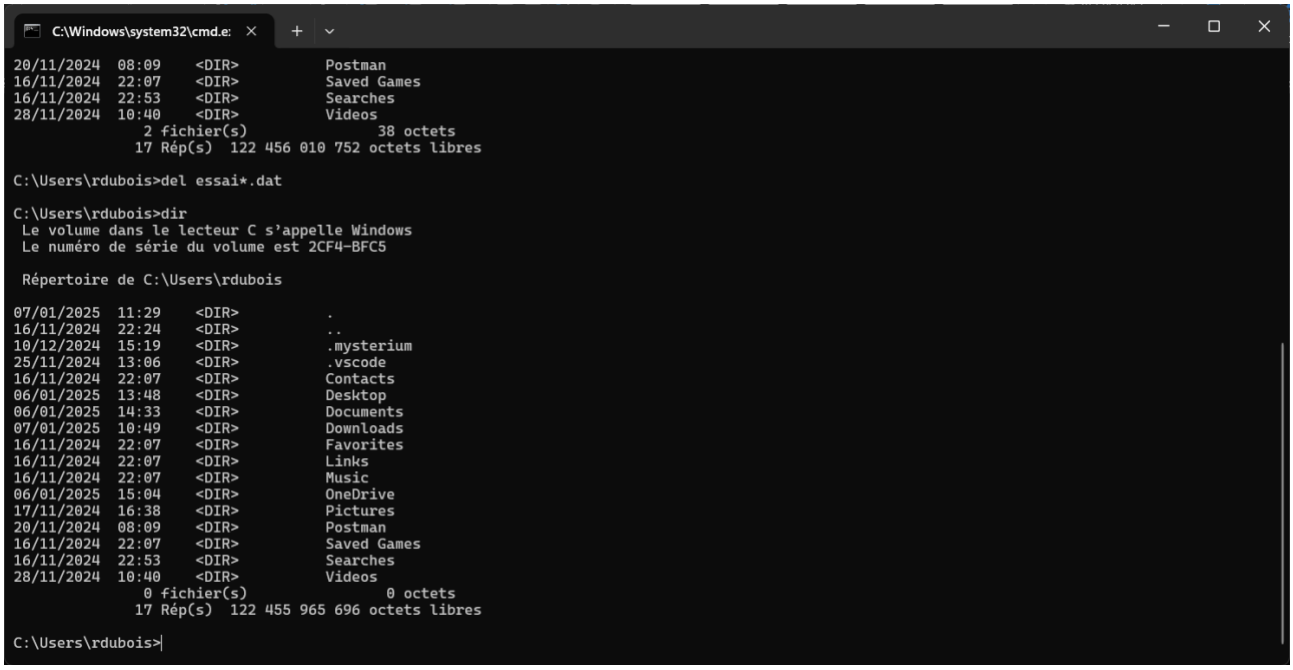
Copiez (copy) le fichier texte.dat vers texte1.dat.



Détruisez (del) en une seule fois tous les fichiers dont le nom commence par "texte".



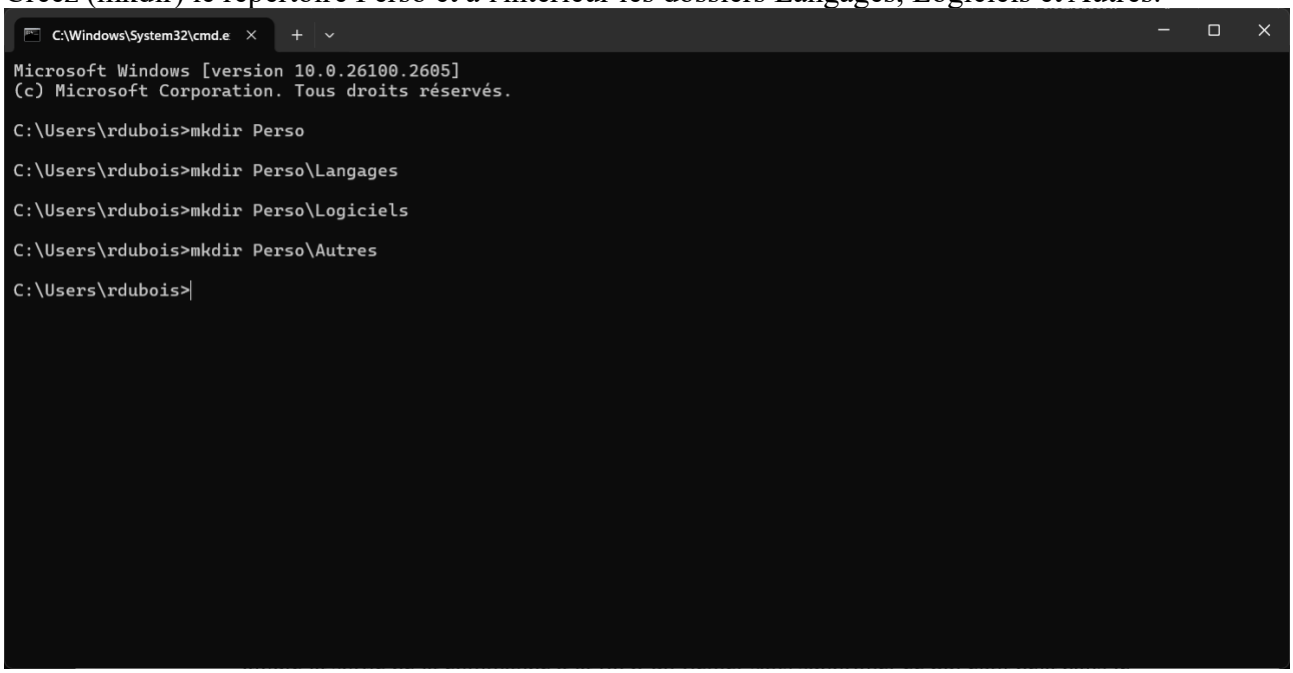
Détruisez (del) tous les fichiers "essai?.dat".



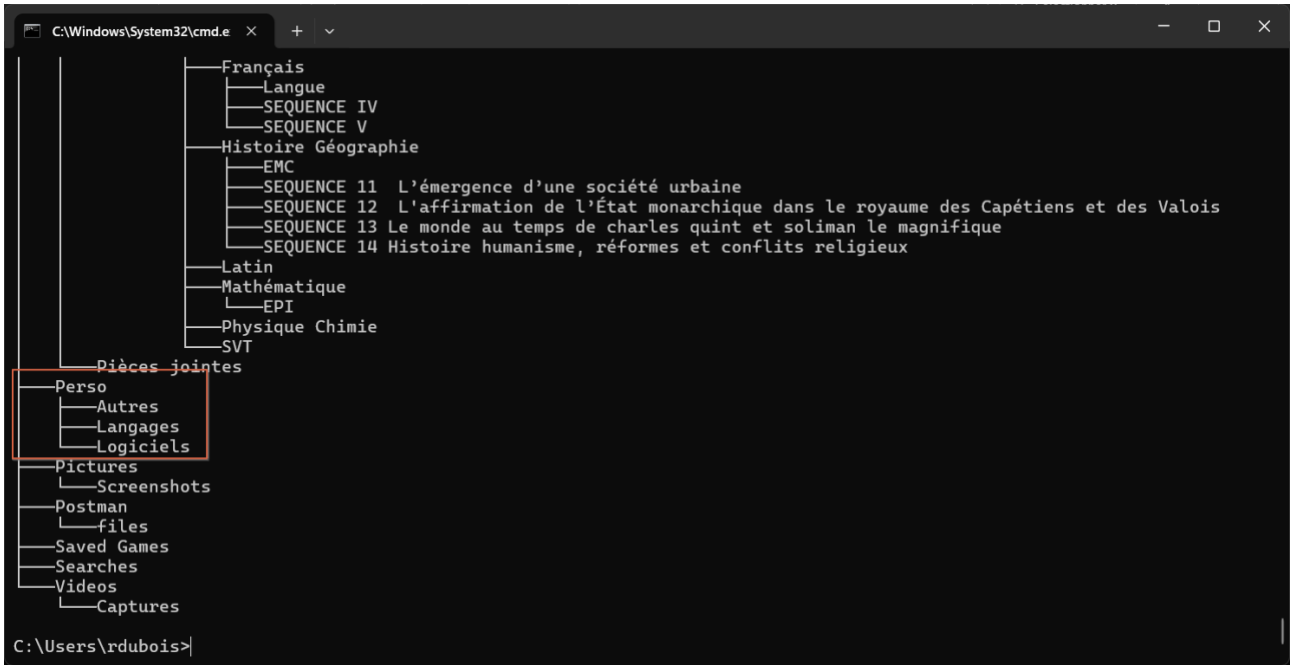
La petite étoile « * » permet de supprimer tous les fichiers commençant par essai et finissant par .dat.

GESTION DES RÉPERTOIRES

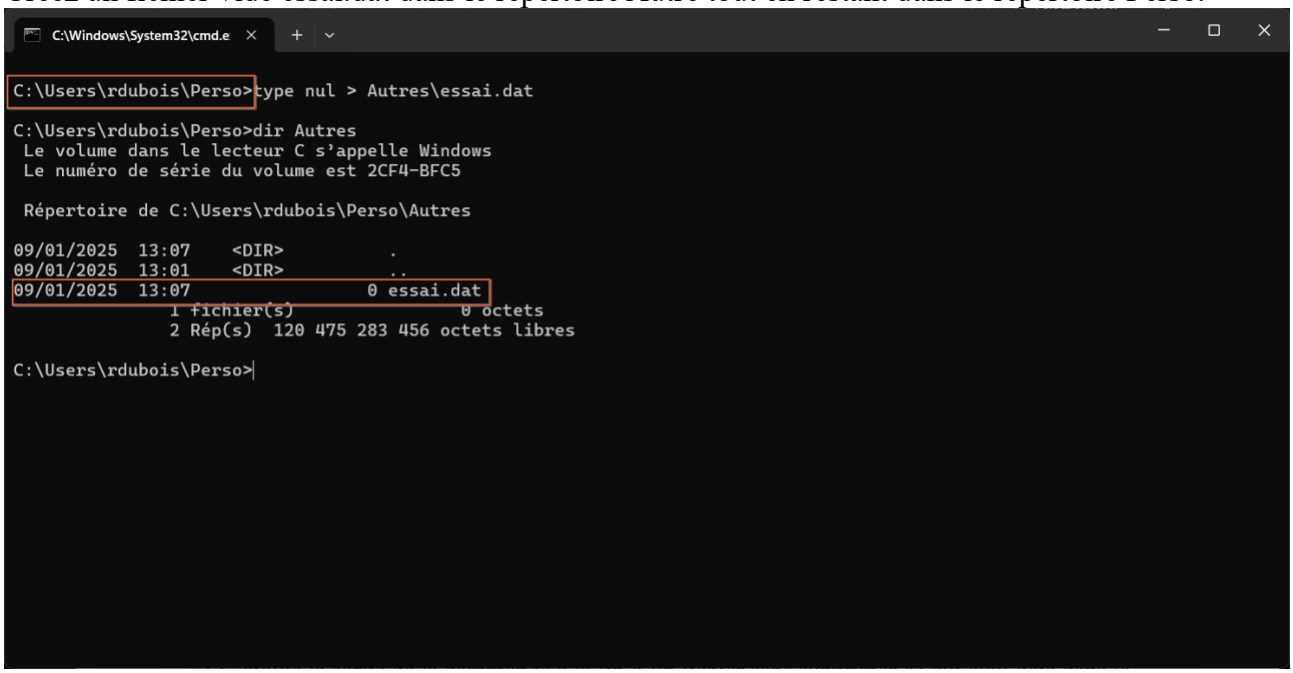
Créez (mkdir) le répertoire Perso et à l'intérieur les dossiers Langages, Logiciels et Autres.



Visualisez (tree) la hiérarchie créée.

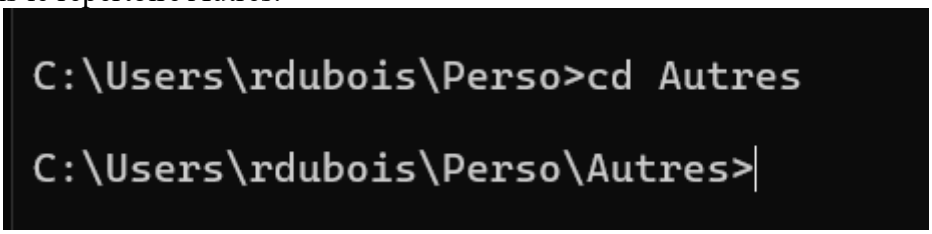


Créez un fichier vide essai.dat dans le répertoire Autre tout en restant dans le répertoire Perso.



La commande type nul a pour rôle d'afficher aucun contenu, ce qui est possible de rediriger cette information avec une redirection dans le bon répertoire, puis en un fichier.

Allez (cd) dans le répertoire Autres.



Le répertoire de travail étant Autres, copiez ce fichier essai.dat sous le répertoire Langages.

```
C:\Users\rdubois\Perso\Autres>copy essai.dat ..\Langages
        1 fichier(s) copié(s).

C:\Users\rdubois\Perso\Autres>dir ..\Langages
 Le volume dans le lecteur C s'appelle Windows
 Le numéro de série du volume est 2CF4-BFC5

 Répertoire de C:\Users\rdubois\Perso\Langages

09/01/2025  13:09    <DIR>          .
09/01/2025  13:01    <DIR>          ..
09/01/2025  13:07                 0 essai.dat
               1 fichier(s)              0 octets
               2 Rép(s)  120 474 009 600 octets libres

C:\Users\rdubois\Perso\Autres>
```

Les ".." permettent de se rendre dans le répertoire précédent

Détruisez (rmdir) le répertoire Perso et tout ce qu'il contient en une seule commande.

```
C:\Users\rdubois\Perso\Autres>cd ..\..

C:\Users\rdubois>rmdir /S /Q Perso
```

⚠ Pour supprimer le fichier, il faut d'abord quitter le dossier avec un « cd .. » (Dans notre cas il a été réalisé 2 fois pour retourner dans le répertoire personnel)
La commande rmdir avec les arguments /S (Supprime tout ce qui a dans le répertoire, que ce soit sous répertoires ou fichiers, sans cet argument, la suppression ne fonctionne pas) /Q (Ne demande pas de confirmation pour supprimer le dossier) permet de supprimer ce dossier et tout ce qu'il contient

LES REDIRECTIONS

Par défaut, MS-DOS lit l'entrée d'une commande saisie au clavier et envoie la sortie sur l'écran. Or il est parfois utile de diriger la sortie d'une commande vers un fichier ou vers l'imprimante.
Par exemple la fonction tree ne possède pas de paramètre permettant d'afficher son résultat page par page, il peut donc être intéressant d'envoyer le résultat de cette commande directement sur l'imprimante ou dans un fichier que l'on visualisera ensuite sous un éditeur.
">" redirige la sortie d'une commande vers un fichier ou vers un périphérique. Exemples :

dir c: > listerep.txt
tree c: > arbre.txt

"<" extrait l'entrée de la commande d'un fichier plutôt que d'un clavier : *sort < listerep.txt*
">>" ajoute la sortie de la commande à la fin d'un fichier sans supprimer ce qui était déjà dans le fichier : *dir c: >> listerep.txt*
Pour créer un fichier vide : *type NUL > vide.txt*

LES FILTRES

Les filtres retraitent des portions de données qui passent par eux
. MS-DOS possède trois commandes :

- more : qui affiche le contenu d'un fichier ou le résultat d'une commande un écran à la fois.
- find : qui recherche des caractères spécifiés dans un fichier ou dans les sorties d'une commande
- sort : qui tri par ordre alphabétique des fichiers et des sorties de commandes.

Exemples d'utilisation :

On veut afficher l'arborescence des répertoires page par page :

tree c: > arbre.txt
more < arbre.txt

On veut afficher uniquement les fichiers d'un répertoire :

dir c:\ > liste.txt
find /v "Windows" < liste.txt > listefic.txt
more < listefic.txt

On veut connaître tous les fichiers du disque dur et les voir par ordre alphabétique.

dir c:\ > listefic.txt
find /v "Windows" < listefic.txt > pasrep.txt
sort < pasrep.txt > fictri.txt
more < fictri.txt

LE TUYAU (pipe)

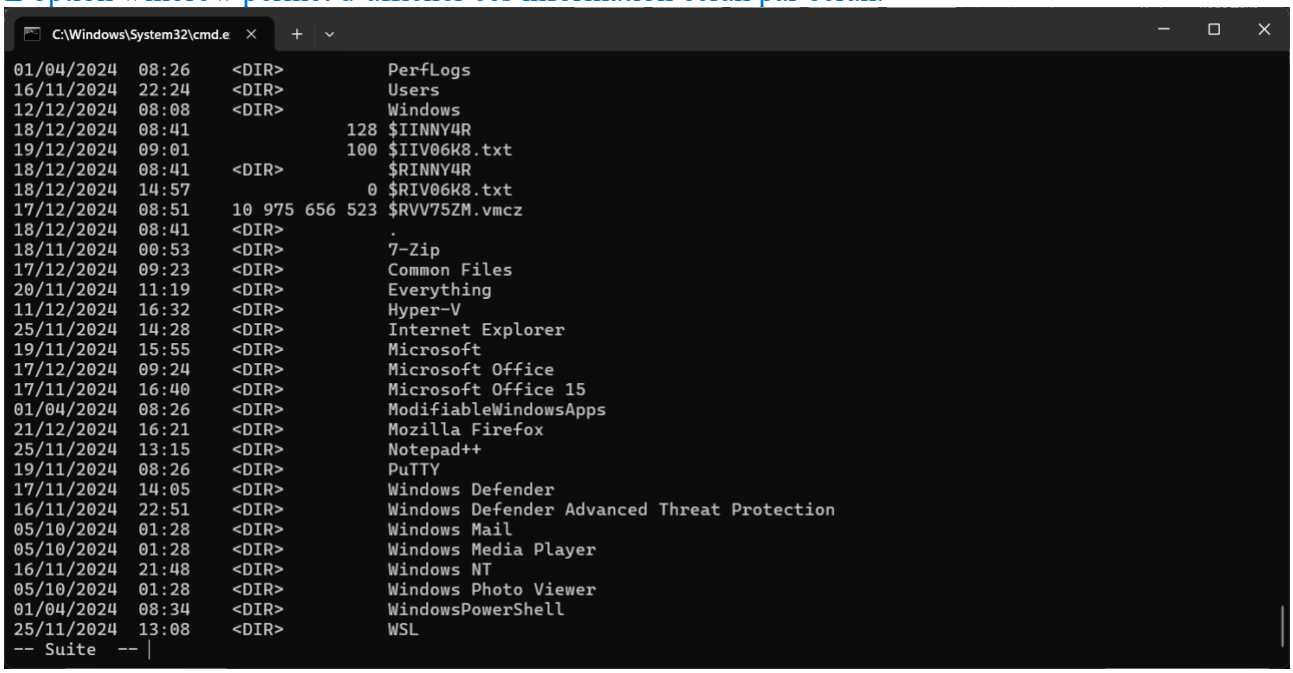
Pour éviter le création de nombreux fichiers intermédiaires on peut faire de la redirection directe entre plusieurs commandes, en utilisant le symbole "|".

tree c:\ | more
dir c:\ | find /v "Windows" | more
find /v "Windows" < listefic.txt | sort | more

EXERCICES

Afficher écran par écran et par ordre alphabétique tous les fichiers du disque dur créés en 2021.

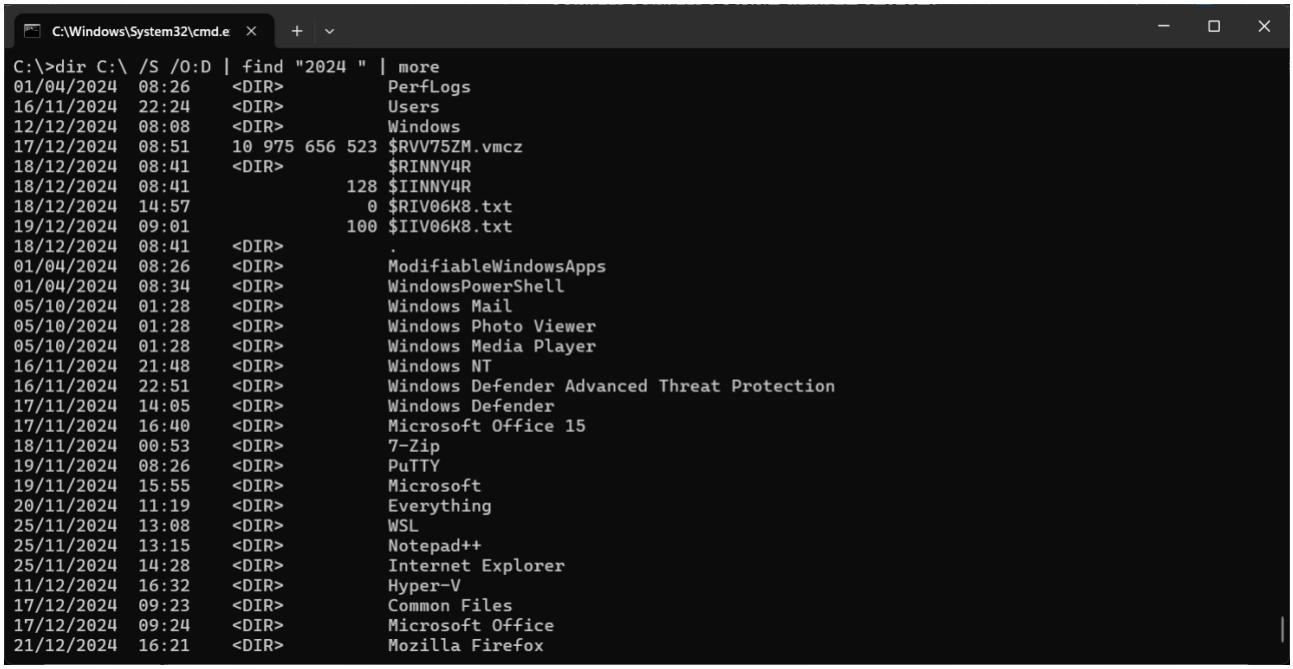
Étant donné que le disque n'a pas de fichiers datant de 2021, on va réaliser cela avec 2024
Pour cela il faut utiliser la commande suivante : dir C:\ | find "2024 " | more
Le dir C:\ /S a pour rôle de lister tous les fichiers du disque dur.
Le find "2024 " a pour rôle d'afficher seulement les fichiers datant de 2024.
L'option « more » permet d'afficher ces information écran par écran.



Pour afficher les fichiers suivants, il suffit d'appuyer sur espace ou entrer

Idem mais les afficher triés par heure (2 solutions : option de la commande dir ou de sort).

Pour les trier par date et heure, il suffit de reprendre la commande précédente et de rajouter l'argument /OD (permettant de trier les fichiers/dossiers par ordre chronologique) sur la commande dir



Idem mais les afficher triés par taille décroissante.

LES FICHIERS DE COMMANDES

Ces fichiers se composent de lignes de commande DOS qui sont lues et exécutées au fur et à mesure. Lorsque toutes ces commandes sont épuisées, le système d'exploitation reprend la main. Ces fichiers de traitement par lot fournissent à l'utilisateur un moyen souple et puissant pour gérer l'utilisation du système. Entre autre, ils évitent la répétition de frappes fastidieuses de commandes. Ces fichiers de traitement par lot sont reconnus par le DOS par leur extension .bat (abréviation de BATCH se traduisant par lot).

Exemple de fichier de commande :

c:\> type exemple.bat
date
time
dir /W

Une fois constitués, ces fichiers sont considérés comme des commandes par DOS. Pour les lancer en exécution il est inutile de préciser leur extension. L'exécution de ces fichiers peut s'interrompre à tout moment par Ctrl-C.

QUELQUES INSTRUCTIONS SPÉCIFIQUES AUX FICHIERS .BAT

L'instruction ECHO

Par défaut, MS-DOS affiche sur l'écran les commandes des fichiers de commandes avant de les exécuter. L'instruction ECHO commande cette visualisation.

ECHO OFF	inhibe cette visualisation mais affiche cette commande à l'écran
ECHO ON	Rétablit cette visualisation (le mode normal)

ECHO	sans argument, affiche à l'écran le mode actuel
ECHO <msg>	dirige le contenu du message vers l'écran
@ECHO OFF	@ inhibe l'affichage de ECHO OFF

L'instruction REM (remarque)
REM <message> agit comme ECHO sauf que REM n'est pris en compte que dans le mode ECHO ON. En fait ECHO s'utilise pour renseigner l'opérateur et REM pour documenter le fichier de commande.

L'instruction PAUSE
L'instruction suspend l'exécution du fichier de commande et affiche "Appuyer sur une touche pour continuer..." Elle sert à donner à l'opérateur le temps de réfléchir, la possibilité de changer un disque, la possibilité d'interrompre l'exécution par Ctrl-C...

L'instruction GOTO
Le label est une chaîne de 1 à 8 caractères alphanumériques. Il apparaît en position d'argument dans l'instruction GOTO. Utilisé comme étiquette sur une ligne de commande, il doit être précédé de ":".
Exemple :

goto FIN
...
:FIN
date

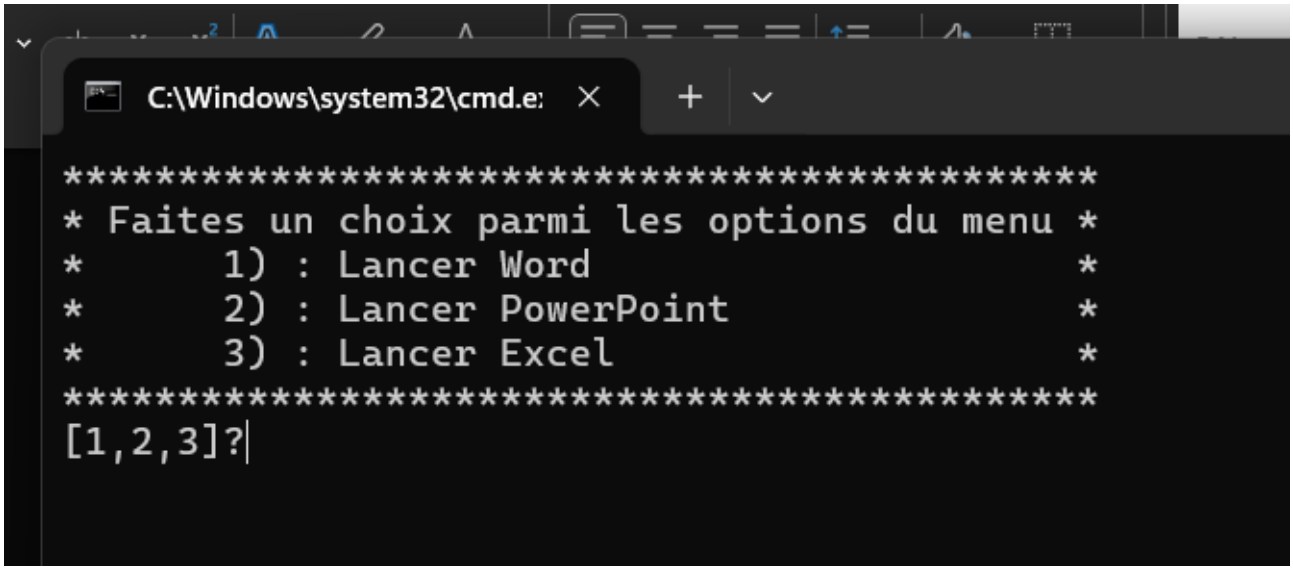
Si le label n'est pas défini en position d'étiquette lors d'un GOTO, l'exécution du fichier est stoppée. Le message "label not found" s'affiche.

LES MENUS
L'instruction CHOICE
Elle permet de choisir parmi des valeurs proposées et de définir une valeur par défaut qui sera prise en compte au bout d'un temps défini. Le choix effectué modifie la variable ERRORLEVEL qui sera testée grâce à l'instruction IF. Exemple :

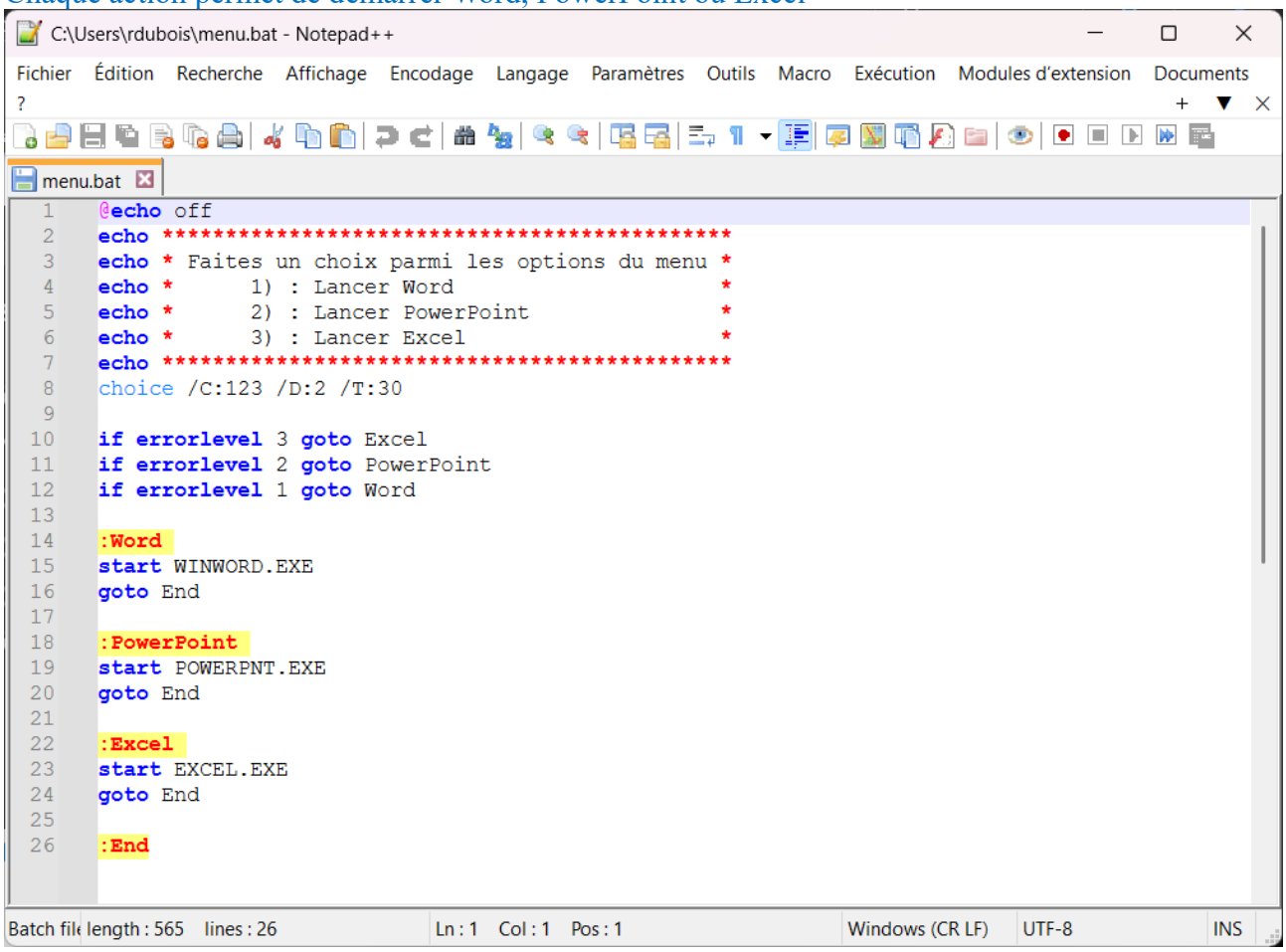
*echo **
echo Faites un choix parmi les options du menu
echo Choix 1 : Lancer Word
echo Choix 2 : Revenir à MS-DOS
*echo **
choice /C:12 /D:2 /T:30

Permet de choisir entre la valeur 1 et la valeur 2. La valeur 2 est choisie par défaut au bout de 30 secondes. (On peut remplacer les : par un espace.)
Exercice : Faire un menu à trois choix et soigner le graphisme de présentation.
Pour cet exercice, j'ai réalisé un menu fonctionnel, permettant d'ouvrir Word, PowerPoint et Excel.
Dans un fichier bat nommer menu.bat

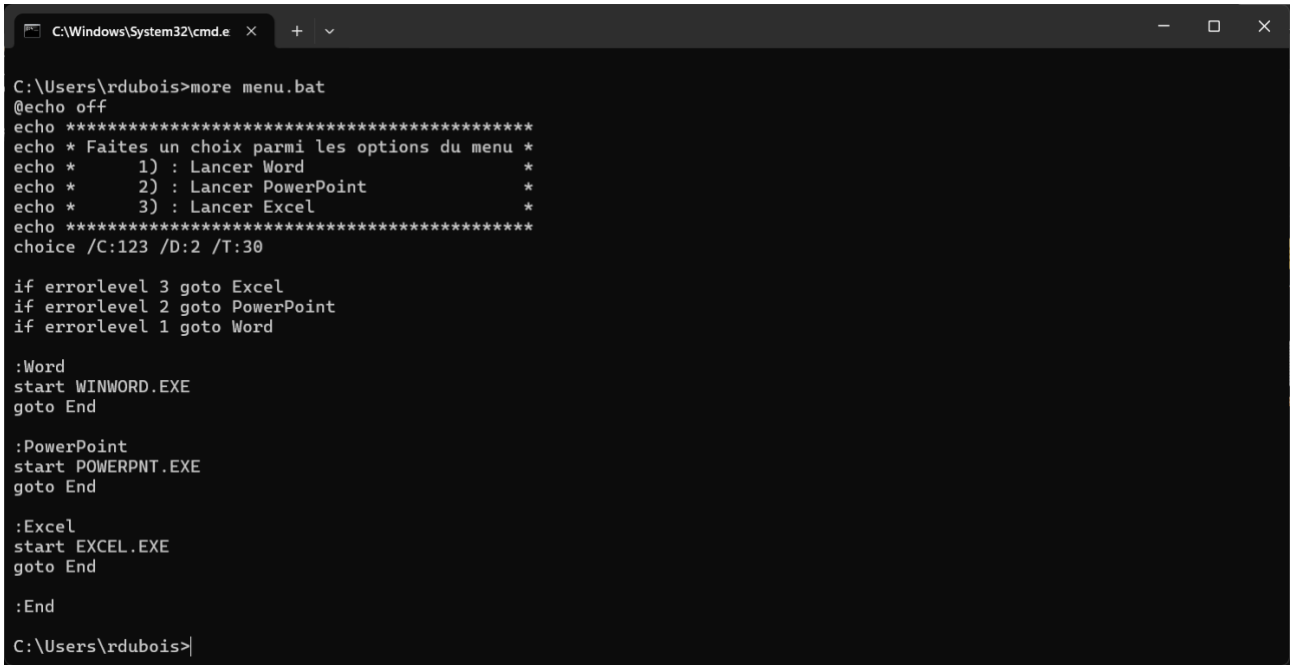




Chaque action permet de démarrer Word, PowerPoint ou Excel



e



Dans ce script, un menu s'affiche grâce aux commandes echo qui vont permettre d'afficher du texte (@echo off permet de masquer certaines informations, comme le répertoire actuel n'étant pas nécessaire pour notre script)

Ensuite la commande « choice » va créer un choix multiple allant de 1 à 3 (et si rien n'est choisi au bout de 30 secondes, l'option 2, Lancer PowerPoint est automatiquement sélectionné)
Une fois le choix effectué, les if errorlevel (numéro choisi) goto (fonction) permettent de lancer une fonction créer au préalable dans le fichier.

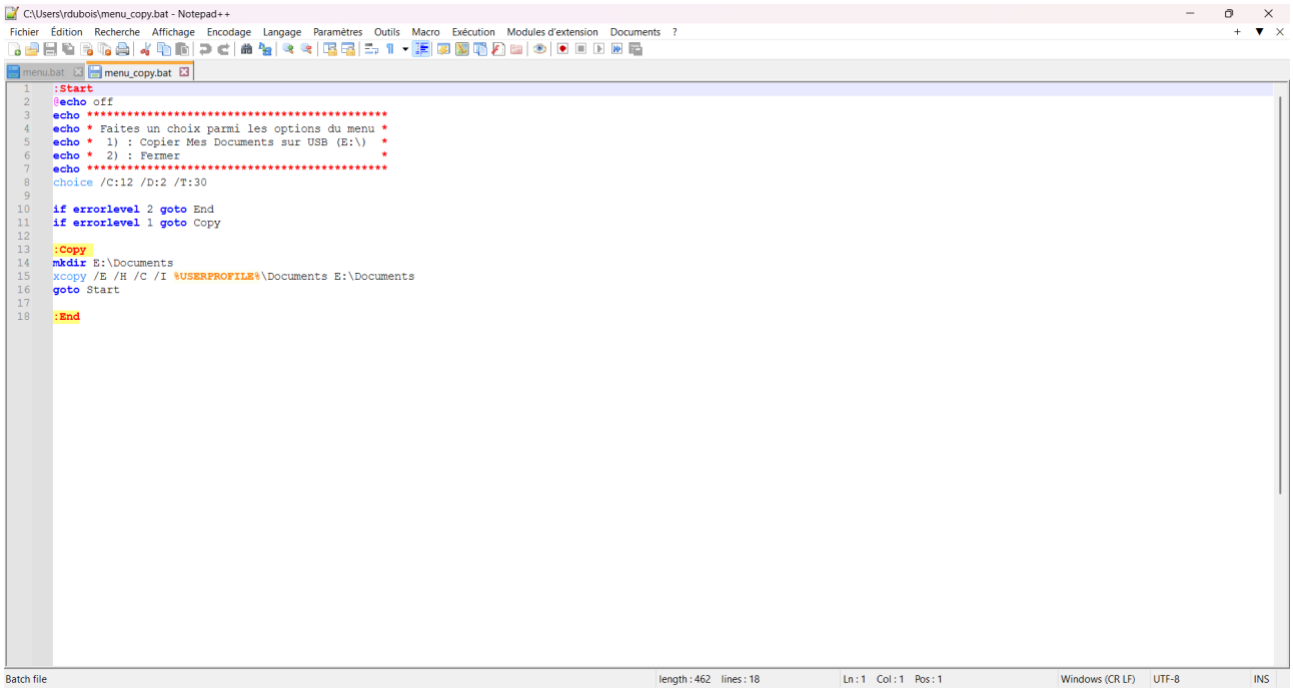
J'ai donc créé 1 étiquette pour chaque action (ayant pour rôle de lancer un des logiciel).

Une fois la fonction exécutée, elle est redirigée vers une autre fonction nommé « :End » qui aura juste le rôle de fermer le script (sans lancer une autre fonction).

L'instruction IF
Elle permet dans le cas d'un menu créé avec l'instruction CHOICE, d'orienter le traitement en fonction du choix de l'utilisateur.
IF ERRORLEVEL <valeur> GOTO <label>
Attention, il s'agit ici d'un test >= et non pas d'un test =. Il faut donc classer les valeurs possibles par ordre décroissant. Pour notre menu :

if errorlevel 2 goto Word
if errorlevel 1 goto Fin
:Word
...
:Fin
...

Exercice : Écrire un fichier de commande permettant de faire plusieurs copies (si l'utilisateur le souhaite) sur des clés (d:) différentes du répertoire Mes Documents.



⚠ La lettre « D » a été remplacée par « E » étant donné que la machine possède déjà 2 disques, 1 en C:\ et un autre sur D:\

Le script est assez similaire à ce qui a été fait pour le menu précèdent. Cependant, une commande particulière a été indiqué : « xcopy » étant un copy avec plus d'options

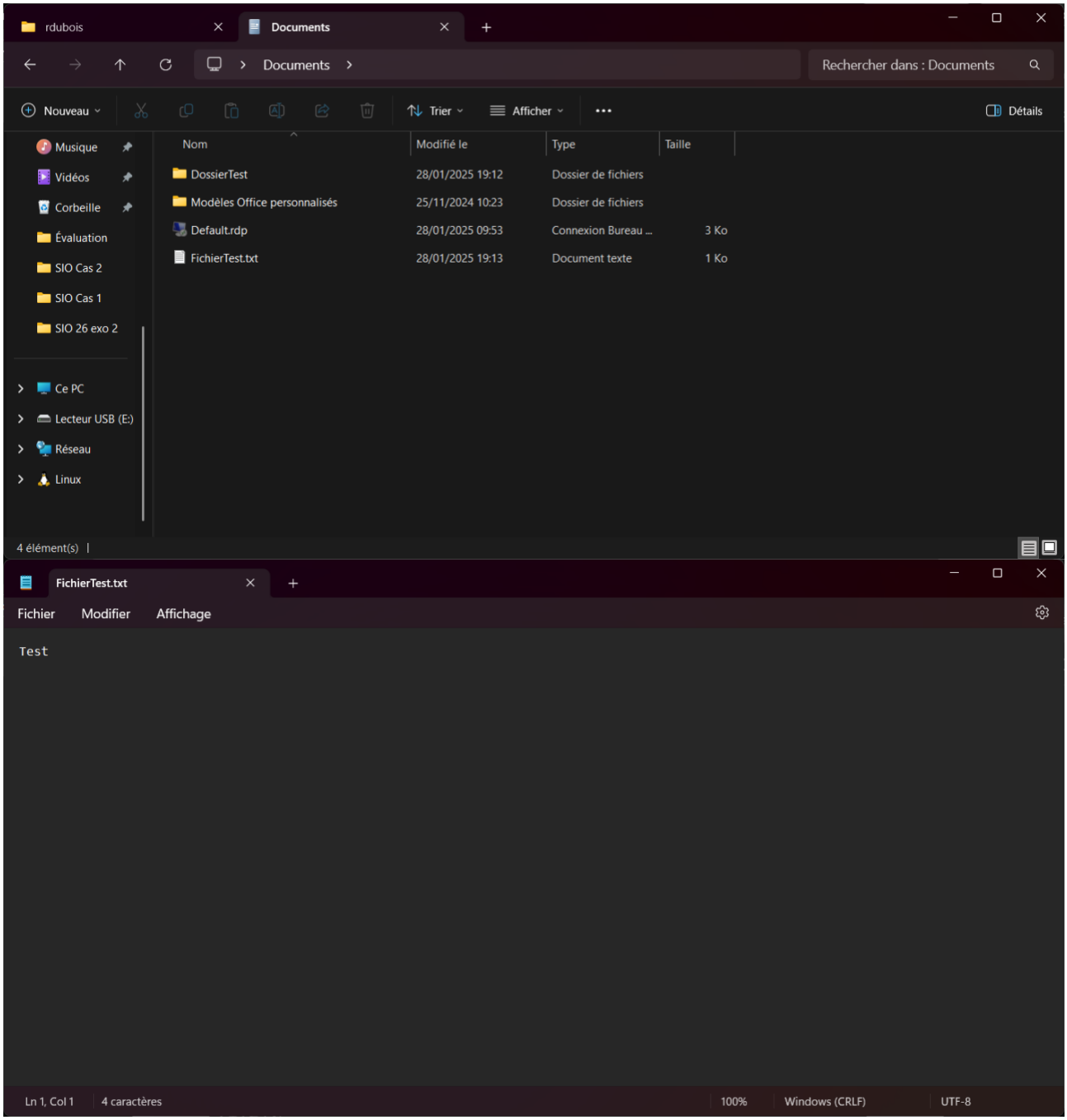
Rôle des différents arguments :

/E : Cet argument a pour rôle de copier tous les répertoires et sous répertoires, même ceux qui sont vide dans le dossier documents.

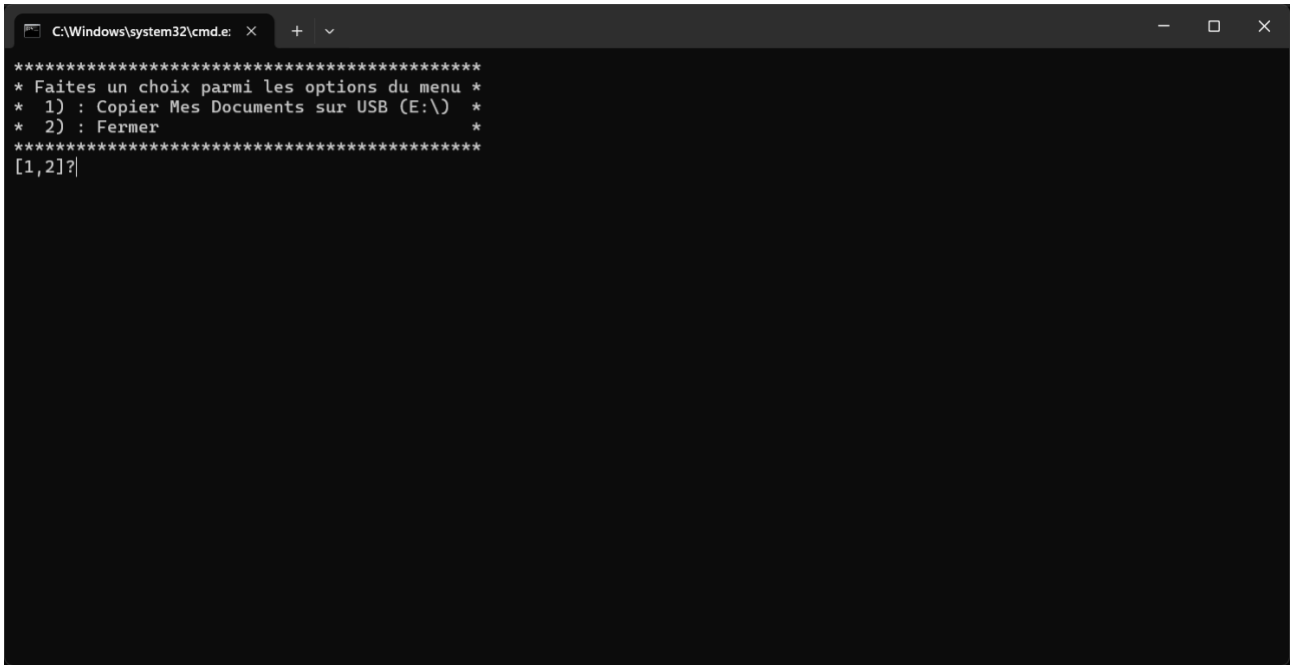
/H : Permet également de copier les fichiers cacher (dans certains cas, la copie ne peut ne pas fonctionner sans cet argument).

/C : Permet de continuer la copie même si une erreur survient.

/I : Permet de faire comprendre à xcopy que la destination est forcément un répertoire et ne pas demander de confirmation lors de la copie des fichiers.



Ceci est les fichiers (dont le contenu de FichierTest.txt) et répertoires présents dans le répertoire Mes Documents avant l'utilisation du script, une clé USB a été inséré dans la machine

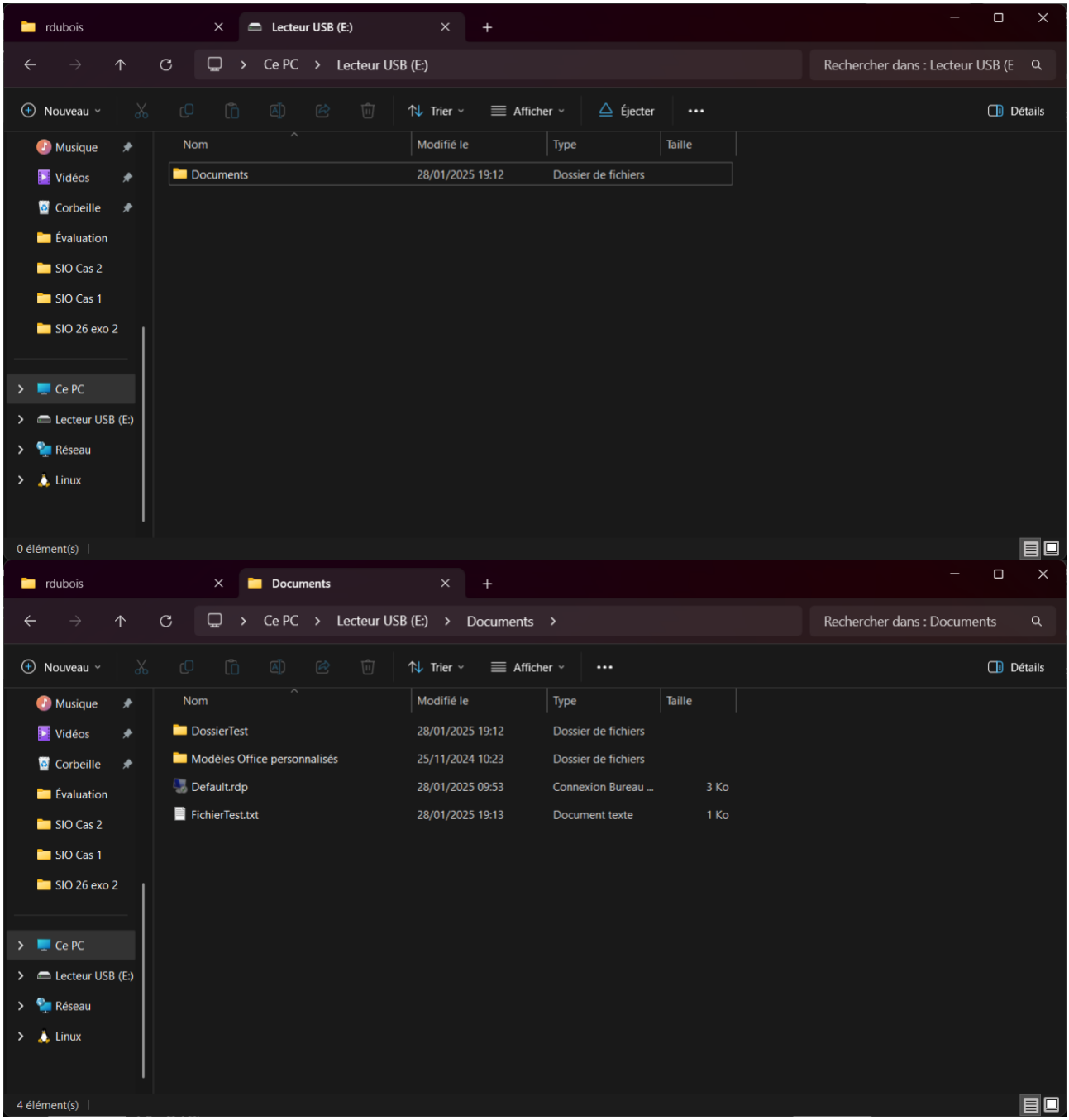


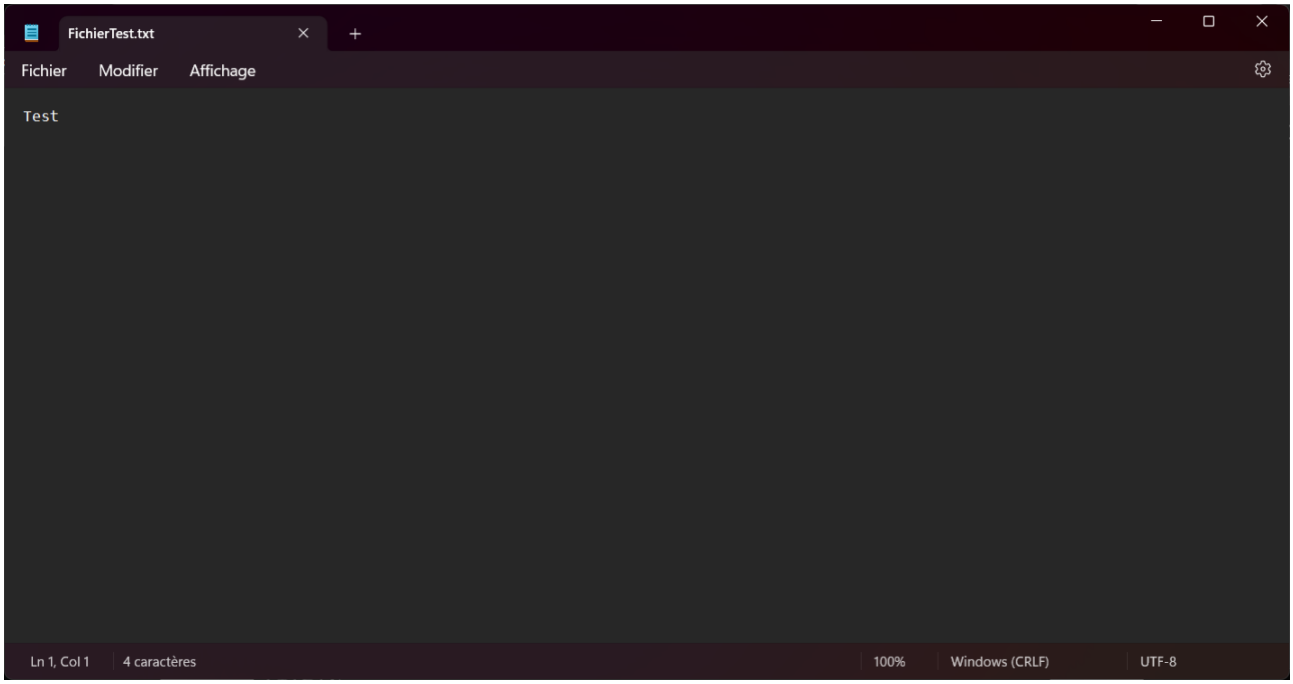
La touche 1 est pressée pour copier les fichiers



Il n'est pas nécessaire de tenir compte des accès refusé, ceci est des dossiers inaccessibles n'ayant pas d'incident sur la continuité du script.

Une fois que le script retourne sur le menu principal, on peut le fermer en appuyant sur 2, et vérifier le contenu de notre clé USB


rdubois
Lecteur USB (E:)
Ce PC › Lecteur USB (E:)
Rechercher dans : Lecteur USB (E
Nouveau
Trier
Afficher
Éjecter
Détails
Musique
Vidéos
Corbeille
Évaluation
SIO Cas 2
SIO Cas 1
SIO 26 exo 2
Ce PC
Lecteur USB (E:)
Réseau
Linux
Nom
Modifié le
Type
Taille
Documents
28/01/2025 19:12
Dossier de fichiers
0 élément(s)
rdubois
Documents
Ce PC › Lecteur USB (E:) › Documents ›
Rechercher dans : Documents
Nouveau
Trier
Afficher
Détails
Musique
Vidéos
Corbeille
Évaluation
SIO Cas 2
SIO Cas 1
SIO 26 exo 2
Ce PC
Lecteur USB (E:)
Réseau
Linux
Nom
Modifié le
Type
Taille
DossierTest
28/01/2025 19:12
Dossier de fichiers
Modèles Office personnalisés
25/11/2024 10:23
Dossier de fichiers
Default.rdp
28/01/2025 09:53
Connexion Bureau ...
3 Ko
FichierTest.txt
28/01/2025 19:13
Document texte
1 Ko
4 élément(s)



On peut donc en conclure que le contenu du dossier Mes Documents a bien été copié avec succès dans notre clé USB

LES PARAMÈTRES

Les commandes incluses dans un fichier de commandes peuvent être paramétrées, c'est-à-dire que les objets de ces commandes sont des paramètres formels. Les paramètres réels seront fournis lors du lancement du fichier batch : c:\> nom-fichier-batch param1 param2

Exemple : *c:\> affichage rouge noir*

Lors de l'appel du fichier, les paramètres réels seront substitués aux paramètres formels à mesure de l'exécution des différentes commandes.

Les paramètres formels s'écrivent %n , où n est un chiffre de 0 à 9 représentant la position du paramètre réel dans la liste des paramètres réels fournie lors du lancement du fichier de commande.

%0 représente donc le nom du fichier de commande lui-même.

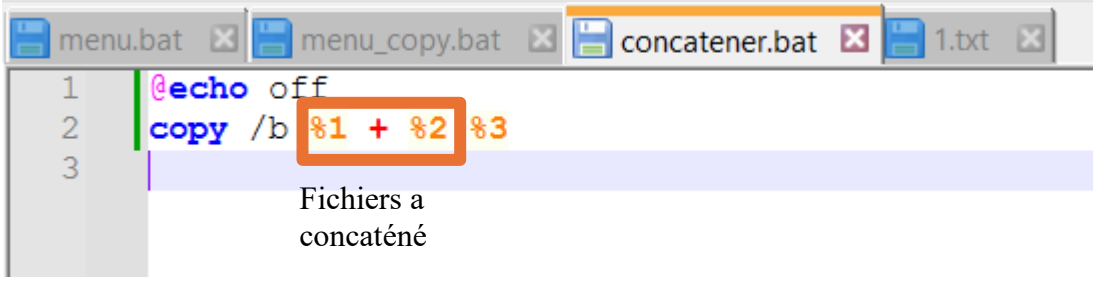
%1 représente la valeur "rouge".

%2 représente la valeur "noir".

Dans le fichier affichage.bat, la ligne de commande est echo %1 affichera "rouge".

Exercice : Construire un fichier de commande concaténant 2 fichiers existants pour en créer un 3e (Voir copy).

Pour concaténer 2 fichiers existant grâce à un fichier bat, il est possible d'utiliser les paramètres permettant de définir les fichiers que nous souhaitons concaténer au lancement du script

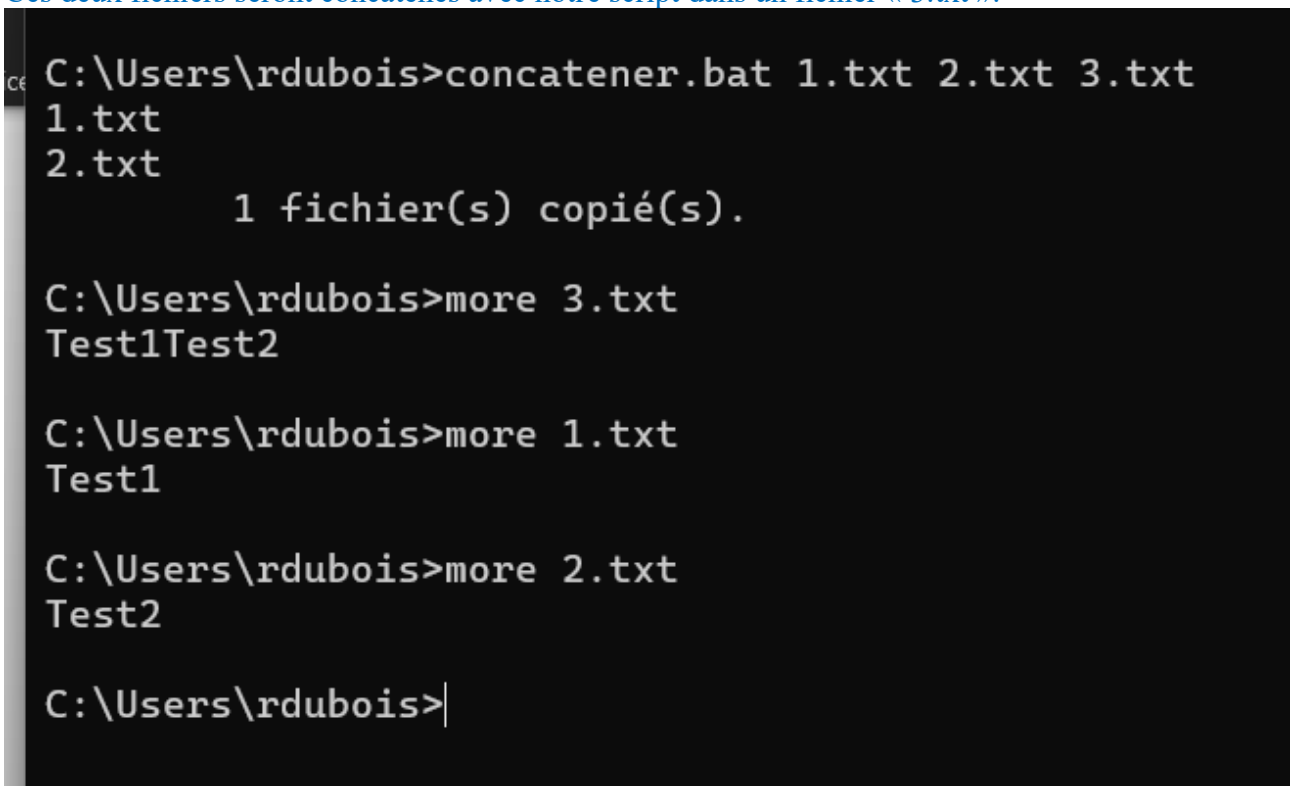


Cette commande va donc concaténer le paramètre 1 avec le 2, puis enfin va le placer dans un fichier

ayant le nom du 3ème argument.
Exemple, avec 2 fichiers :

Un fichier : 1.txt contenant le texte « Test1 »
Un deuxième fichier : 2.txt contenant le texte « Test2 »

Ces deux fichiers seront concaténés avec notre script dans un fichier « 3.txt ».



On peut donc observer que notre script fonctionne et à bien concaténer les deux fichiers dans un fichier 3.txt.

LES BOUCLES
L'instruction IF
L'instruction conditionnelle se présente sous la forme :

IF [NOT] <condition> <commande1> [ELSE <commande2>]

Les comparants utilisables sont :

- EQU - égal à
- NEQ - différent de
- LSS - inférieur à
- LEQ - inférieur ou égal à
- GTR - supérieur à
- GEQ - supérieur ou égal à

Ou bien on peut tester l'existence d'un fichier. Exemple : *if not exist test.jpg echo fichier manquant*

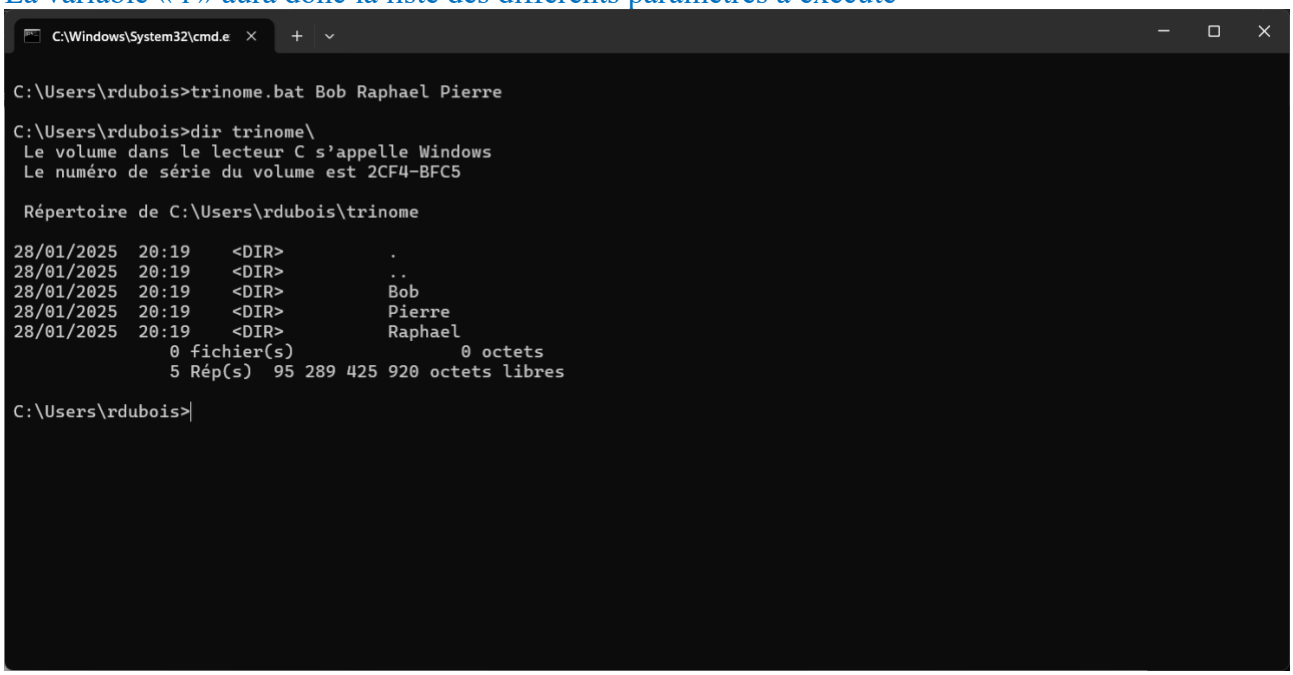
L'instruction FOR
C'est une boucle d'itération qui se présente sous la forme :

FOR %%<variable> IN (<liste>) DO <commande>

Exercice : créer un sous-répertoire "trinome" contenant 3 sous-répertoires portant le nom de chaque participant du trinôme entrés comme paramètres.
Pour réaliser cela, nous devrons faire un script en .bat devant créer 1 dossier trinôme avec mkdir, puis il faudra faire une boucle s'exécutant

```
1  @echo off
2  mkdir trinome
3  for %%i in (%1 %2 %3) do (
4      mkdir trinome\%%i
5  )
```

La variable « i » aura donc la liste des différents paramètres a exécuté



Les paramètres sont bien en compte et notre boucle aussi ce qui nous permet de créer les différents sous dossier de notre choix grâce à un simple script .bat

L'instruction SHIFT
Cette instruction permet d'utiliser plus de 9 paramètres réels par simple décalage.
Après chaque instruction SHIFT le paramètre réel 0 disparaît, le paramètre réel 1 devient 0, etc. Le paramètre 10 qui ne pouvait pas être atteint devient 9 donc disponible et ainsi de suite.
Exercice : Afficher le contenu des répertoires donnés en paramètre sans connaître leur nombre.

```
1  @echo off
2
3  :loop
4  if "%1"=="" goto end
5  dir %1
6  shift
7  goto loop
8
9  :end
```

Si le premier paramètre (%1) est vide, cela signifie qu'il n'y a plus de paramètres à traiter

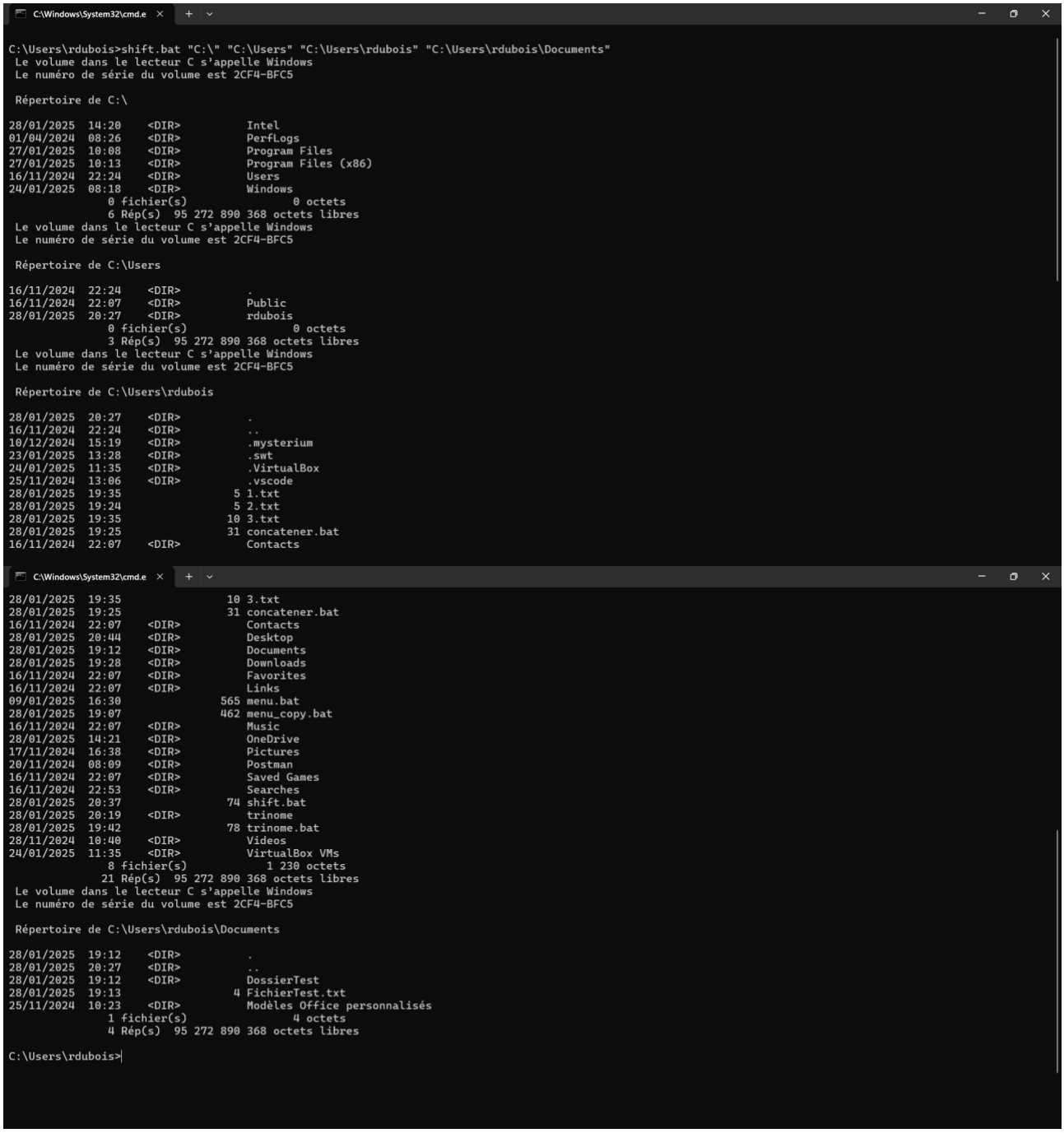
Le script suivant contient une boucle avec un shift ayant pour rôle de décaler le paramètre. Le if "%1"=="" goto end a lui pour rôle lors que les paramètres sont vides (qu'ils ont été tous affichés) de s'arrêter en allant sur l'étiquette end.

Exemple avec 2 répertoires (C:\ et C:\Users) :

```
C:\Windows\System32\cmd.e

C:\Users\rdubois>shift.bat "C:\" "C:\Users"
 Le volume dans le lecteur C s'appelle Windows
 Le numéro de série du volume est 2CF4-BFC5

 Répertoire de C:\

28/01/2025  14:20    <DIR>          Intel
01/04/2024  08:26    <DIR>          PerfLogs
27/01/2025  10:08    <DIR>          Program Files
27/01/2025  10:13    <DIR>          Program Files (x86)
16/11/2024  22:24    <DIR>          Users
24/01/2025  08:18    <DIR>          Windows
               0 fichier(s)                0 octets
               6 Rép(s)  95 273 267 200 octets libres
 Le volume dans le lecteur C s'appelle Windows
 Le numéro de série du volume est 2CF4-BFC5

 Répertoire de C:\Users

16/11/2024  22:24    <DIR>          .
16/11/2024  22:07    <DIR>          Public
28/01/2025  20:27    <DIR>          rdubois
               0 fichier(s)                0 octets
               3 Rép(s)  95 273 267 200 octets libres

C:\Users\rdubois>
```

Les deux répertoires sont affichés avec succès.

Exemple avec 4 répertoires (C:\ C:\Users C:\Users\rdubois C:\Users\rdubois\Documents)



En conclusion, tous les répertoires sont affichés avec succès