

# TSSR - Compte rendu TP8: Scripts Batch Sous Windows

Jean-Paul MELISSE

07 Février 2025

## 1 Introduction

L'objectif de ce TP est de nous familiariser avec les scripts Batch sous Windows. Pour cela, on va créer une série de 8 petits scripts sur une machine virtuelle. Ce petit document va nous décrire les étapes à suivre.

## 2 Créer un dossier de travail

On choisit un emplacement de travail où on va garder tous les fichiers installés et la configuration de la machine Windows. On décide de sauvegarder dans le chemin : "C:\Users\Jean-Paul\Desktop\tp8\", comme montre la Figure 1.

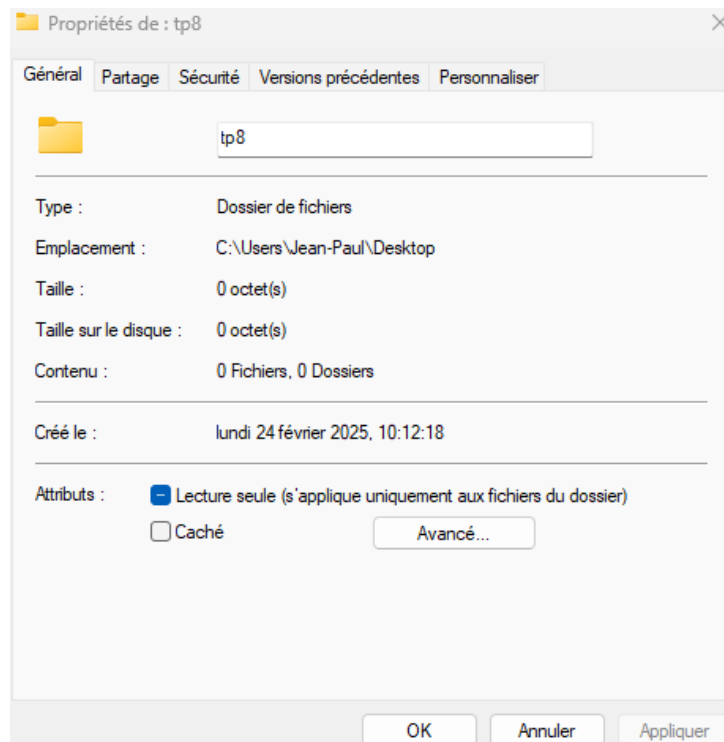


Figure 1

### 3 Télécharger Windows

Dans ce TP, on aura besoin d'un système d'exploitation Windows. On va donc le télécharger via le lien <https://www.microsoft.com/en-us/software-download/windows11>. On choisit "Download Windows 11 Disk Image (ISO) for x64 devices" et l'option "Windows 11 (multi-edition ISO for x64 devices)" et on clique sur le bouton "Download Now". On choisit la langue "English (United States)" et on clique sur "Confirm". Enfin, on clique sur "64-bit Download" pour le télécharger. On le sauvegarde dans le dossier tp8 sous le format .iso (Voir la Figure 2).

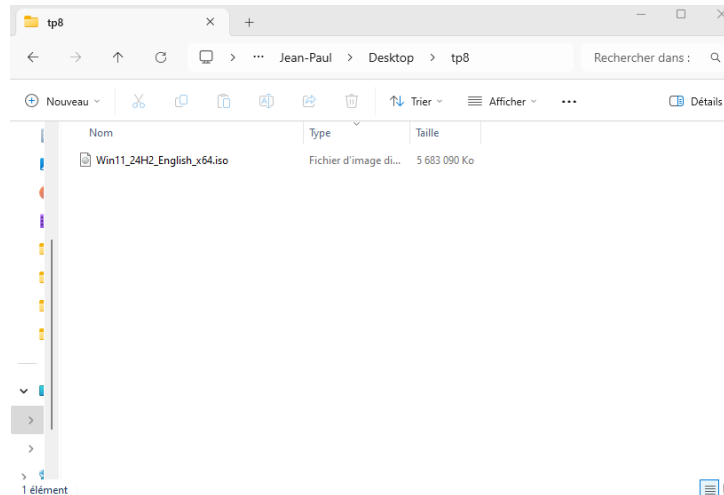


Figure 2

### 4 Créer la machine virtuelle tp8-windows

On ouvre VirtualBox. On clique sur "Nouveau" pour créer une nouvelle machine virtuelle. On met "tp8-windows" comme nom de la machine virtuelle. On choisit son emplacement de sauvegarde et on importe le CD contenant le système d'exploitation Windows. En détectant le CD, le type "Microsoft Windows", le subtype "" et la version "Windows 11 (64 bits)" sont automatiquement mis. On coche la case "Skip Unattended Installation" et on clique sur "Suivant". On lui donne une mémoire vive (RAM) de 4096 Mo et 2 processeurs, et on clique sur "Suivant". On lui donne un disque dur de 80 Go et on clique sur "Suivant". On regarde bien le récapitulatif et on clique sur "Finish" (Voir la Figure 3).

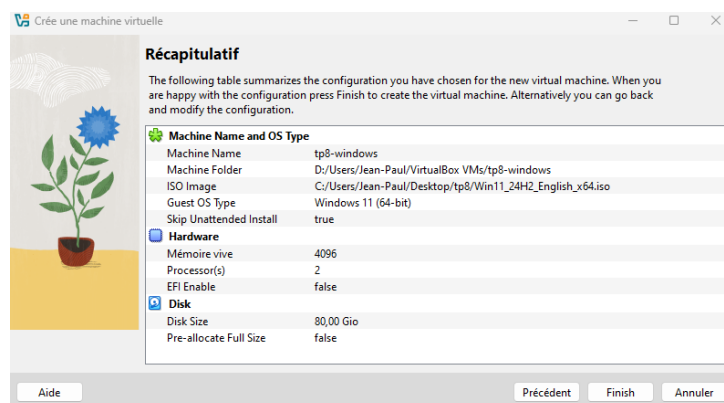


Figure 3

Après avoir créé notre machine virtuelle Windows, on va isoler notre machine. Sur VirtualBox on clique sur "tp8-windows" et sur "Configurations". On clique sur "Réseau" et on change le mode d'accès réseau à "Réseau interne", puis on clique sur "OK". Puis on clique sur "Démarrer" pour l'allumer.

## 5 Installer Windows

Après avoir démarré la machine virtuelle, on clique sur n'importe quelle touche pour passer sur le bootable du CD. On choisit pour le paramètre "Language to install: English (United States)" et on choisit pour l'option "Time and currency format: French (France)". On clique sur "Next". Pour le clavier (Keyboard or input method), on choisit "French (Legacy, AZERTY)" et on clique sur "Next". On coche les cases "Install Windows 11" et "I agree everything will be deleted", et on clique sur "Next". On clique sur "I don't have a product key" pour installer Windows sans clé de produit. On choisit d'installer "Windows 11 Pro" et on clique sur "Next" (cf Figure 4).

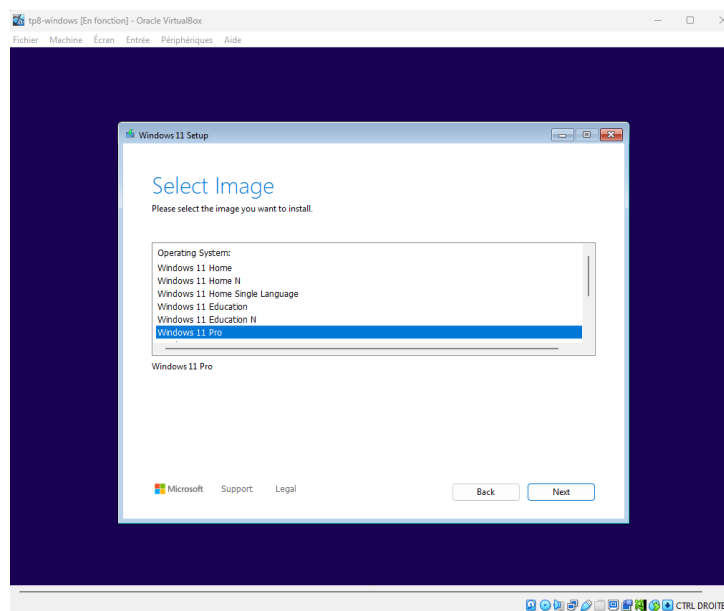


Figure 4

On clique sur "Accept". On choisit le disque "Disk 0 Unallocated Space" pour l'emplacement d'installation et on clique sur "Next". Enfin, on clique sur "Install" pour démarrer l'installation de Windows 11. Après que la machine ait démarré plusieurs fois et qu'on voie la page pour indiquer le pays, on va faire une manipulation afin de pouvoir configurer l'installation finale de Windows sans Internet. Pour ce faire, on appuie sur "Shift + F10".

Cela va faire apparaître le prompt. Sur le prompt, on tape la commande : "oobe\bypassNRO", puis on tape sur "Entrée". La machine va de nouveau redémarrer et on retrouve la même page du choix du pays. On choisit "France" et on clique sur "Next". Pour le clavier, on choisit "French (Legacy, AZERTY)" et on clique sur "Yes". On n'a pas de second clavier, donc on clique sur "Skip". Là, la machine va vouloir se connecter à l'Internet, mais elle n'est connectée qu'au réseau interne (pas Internet). On clique donc sur le bouton "I don't have Internet" (ce bouton est apparu grâce à la commande oobe tapée plus haut). On donne le nom de l'utilisateur "user" et on clique sur "Next". On ne donne pas de mot de passe (on laisse le champ vide) et on clique sur "Next". On clique sur "No" et "Accept", et on répète cette action

1 fois. On choisit l'option "Required only" et on clique sur "Accept". On choisit "No" et on clique sur "Accept", et on répète cette action 2 fois. On laisse Windows finaliser son installation.

Note : sous Windows, nous éditerons tous nos fichiers avec l'éditeur de texte Notepad.

Note : sous Windows, nous testerons tous nos scripts sous cmd.

## 6 Les sorties en Batch

Par la suite du TP, on se concentre sur les scripts Batch. On crée un nouveau dossier "tp8-jean-paul-melisse" qui va contenir tous les scripts Bash. Puis on se rend dans ce dossier.

Les entrées-sorties désignent les échanges de données entre le processeur (CPU) et les périphériques. Les entrées sont les données envoyées par les périphériques d'entrée (clavier, scanner, . . . ) et lues par le CPU. Les sorties sont les données reçues par les périphériques de sortie (écran, imprimante, . . . ) et écrites par le CPU. Dans cette section, nous nous intéressons aux sorties et à écrire à l'écran. Pour écrire à l'écran, nous utilisons la commande echo. Pour en savoir plus sur l'utilisation de cette commande, tapez "echo /?". On va créer un script 01-write.bat qui écrit hello world à l'écran. Ce script doit retourner 0. Pour cela, on ouvre l'application Notepad. Dans ce nouveau fichier, on écrit la ligne : echo off pour éviter que batch affiche le contenu du script sur la console. Puis on ajoute la ligne suivante : "echo hello world", comme montré sur la Figure 5. Puis on ajoute aussi la ligne : "exit /b 0"

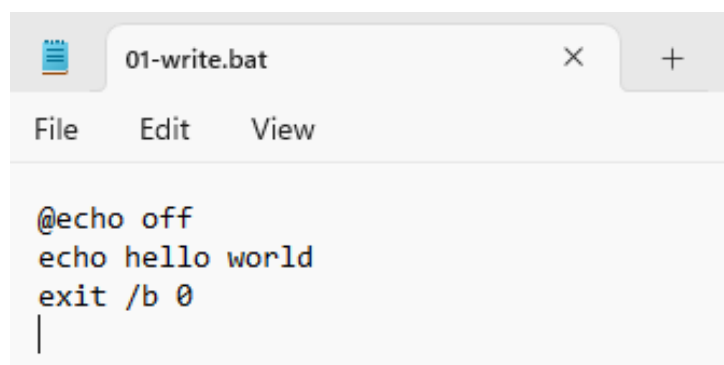


Figure 5

On sauvegarde en utilisant "Ctrl+S" et on enregistre le fichier sous le nom "01-write.bat" dans le dossier tp8. On peut vérifier l'exécution du script en ouvrant l'invite de commande. On appuie sur "Win + R", ce qui lance le programme "RUN". Dans le programme RUN, on tape "cmd" et on clique sur "OK". Sur l'invite de commande, on change de dossier en tapant la commande "cd Documents\tp8-jean-paul-melisse". Puis, on tape la commande : "01-write.bat" (Voir la Figure 6).

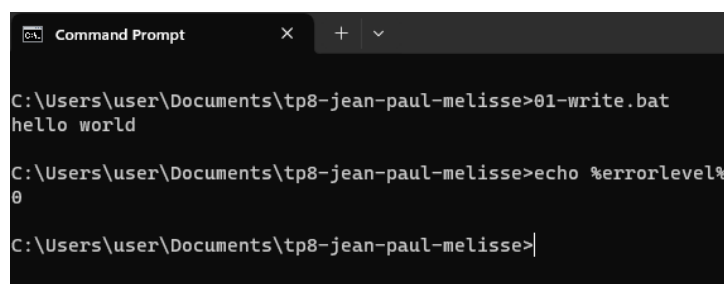


Figure 6

## 7 Les variables en Batch

Les variables permettent d'affecter une valeur à un nom. Pour créer une variable, nous utilisons la commande `set`, exemple : `set name=value` . Pour utiliser une variable créée, nous utilisons le caractère `%`, exemple : `%name%` . On va créer un script `02-variable.bat` qui crée la variable `first_name` en lui affectant la valeur `samson` et qui l'écrit à l'écran. Ce script doit retourner 0. Pour cela, on ouvre l'application Notepad. Dans ce nouveau fichier, on écrit les lignes suivantes :

```
1 @echo off
2 set first_name = "samson"
3 echo %first_name%
4 exit /b 0
```

On sauvegarde en utilisant "Ctrl+S" et on enregistre le fichier sous le nom "`02-variable.bat`" dans le dossier `tp8`. On peut voir l'étape précédente sur la Figure 7. On peut vérifier l'exécution du script en ouvrant l'invite de commande. On appuie sur "Win + R", ce qui lance le programme "RUN". Dans le programme RUN, on tape "`cmd`" et on clique sur "OK". Sur l'invite de commande, on change de dossier en tapant la commande "`cd Documents\tp8-jean-paul-melisse`". Puis, on tape la commande : "`02-variable.bat`" (voir la Figure 8).



Figure 7

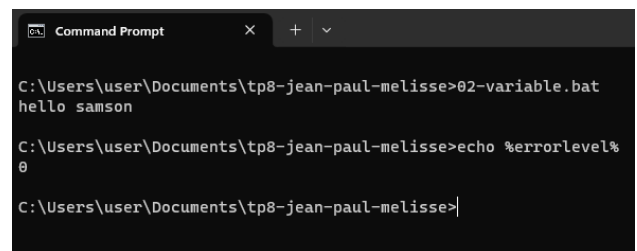


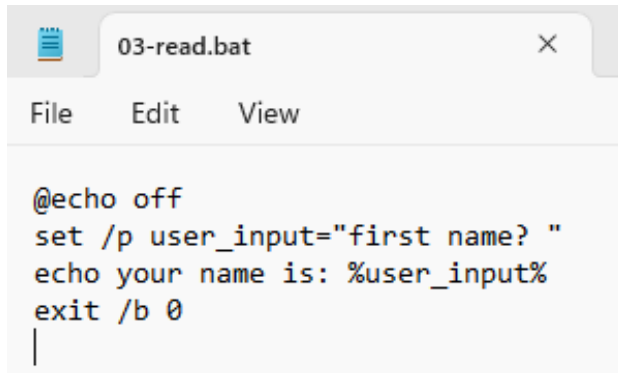
Figure 8

## 8 Les entrées en Batch

Dans cette section, nous nous intéressons aux entrées et à lire au clavier. Pour lire au clavier, nous utilisons la commande "`set`" avec une option spécifique : "`set /p`". On va créer un script "`03-read.bat`" qui lit au clavier une chaîne de caractères représentant un prénom, l'écrit à l'écran. Ce script doit retourner 0. On ouvre l'application Notepad. Dans ce nouveau fichier, on écrit les lignes suivantes :

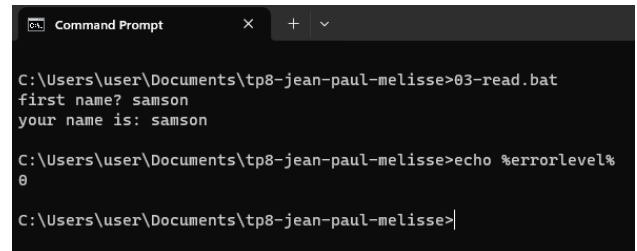
```
1 @echo off
2 set /p user_input="first name? "
3 echo "your name is: " %user_input%
4 exit /b 0
```

On sauvegarde en utilisant "Ctrl+S" et on enregistre le fichier sous le nom "03-read.bat" dans le dossier tp8. On peut voir l'étape précédente sur la Figure 9. On peut vérifier l'exécution du script en ouvrant l'invite de commande. On appuie sur "Win + R", ce qui lance le programme "RUN". Dans le programme RUN, on tape "cmd" et on clique sur "OK". Sur l'invite de commande, on change de dossier en tapant la commande "cd Documents\tp8-jean-paul-melisse". Puis, on tape la commande : "03-read.bat" (voir la Figure 10).



```
@echo off
set /p user_input="first name? "
echo your name is: %user_input%
exit /b 0
```

Figure 9



```
C:\Users\user\Documents\tp8-jean-paul-melisse>03-read.bat
first name? samson
your name is: samson

C:\Users\user\Documents\tp8-jean-paul-melisse>echo %errorlevel%
0

C:\Users\user\Documents\tp8-jean-paul-melisse>
```

Figure 10

## 9 Les conditions en Batch

Bien que les instructions d'un script soient habituellement exécutées de façon séquentielle, il est possible d'en exécuter à l'intérieur d'une structure conditionnelle.

- Syntaxe :

```
1 if CONDITION-1 (
2     STATEMENT-1
3 ) else (
4     STATEMENT-2
5 )
```

On va créer un script "04-grade.bat" qui lit au clavier un entier représentant une note sur 20, écrit que c'est un échec si la note est entre 0 et 9 et écrit que c'est un succès si la note est entre 10 et 20. Si la note est en dehors de ces deux plages, le script doit écrire un message d'erreur et retourner 1 au lieu de 0. Si la note ne peut pas être convertie vers un entier, le script doit écrire un message d'erreur et retourner 1 au lieu de 0. On ouvre l'application Notepad. Dans ce nouveau fichier, on écrit les lignes suivantes :

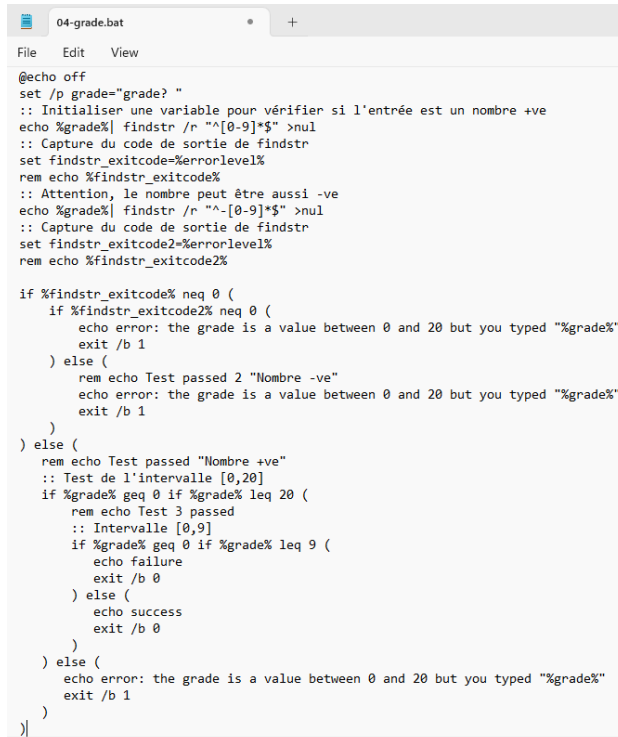
```
1 @echo off
2 set /p grade="grade? "
3
4 echo %grade%|findstr /r "[0-9]*$" >nul
5 Set findstr_exitcode=%errorlevel%
```

```

6
7 echo %grade%|findstr /r "[0-9]*$" >nul
8 Set findstr_exitcode2=%errorlevel
9
10 if %findstr_exitcode% neq 0 (
11     if %findstr_exitcode2% neq 0 (
12         echo error: the grade is a value between 0 and 20 but you typed "%grade%"
13         exit /b 1
14     ) else (
15         rem echo Nombre -ve
16         echo error: the grade is a value between 0 and 20 but you typed "%grade%"
17         exit /b 1
18     )
19 ) else (
20     if %grade% geq 0 if %grade% leq 20 (
21         rem echo Intervalle [0,20]
22         if %grade% geq 0 if %grade% leq 9 (
23             echo failure
24             exit /b 0
25         ) else (
26             echo success
27             exit /b 1
28         )
29     ) else (
30         echo error: the grade is a value between 0 and 20 but you typed "%grade%"
31         exit /b 1
32     )
33 )

```

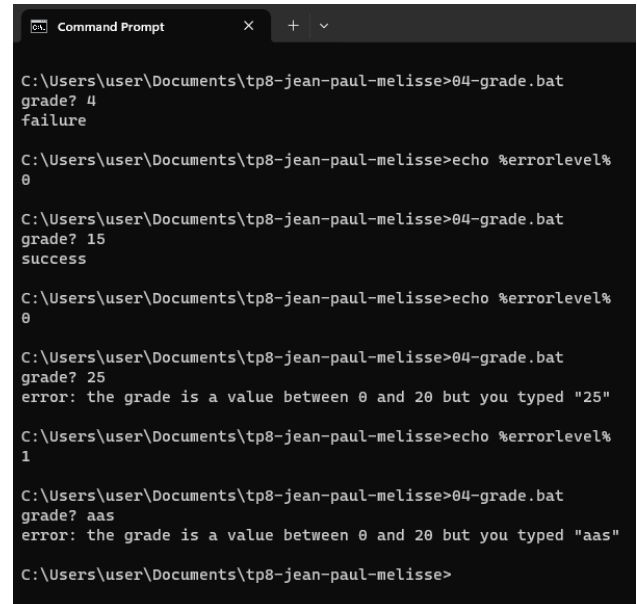
On sauvegarde en utilisant "Ctrl+S" et on enregistre le fichier sous le nom "04-grade.bat" dans le dossier tp8. On peut voir l'étape précédente sur la Figure 11. On peut vérifier l'exécution du script en ouvrant l'invite de commande. On appuie sur "Win + R", ce qui lance le programme "RUN". Dans le programme RUN, on tape "cmd" et on clique sur "OK". Sur l'invite de commande, on change de dossier en tapant la commande "cd Documents\tp8-jean-paul-melisse". Puis, on tape la commande : "04-grade.bat" (voir la Figure 12).



```
@echo off
set /p grade="grade? "
:: Initialiser une variable pour vérifier si l'entrée est un nombre +ve
echo %grade% | findstr /r "^[0-9]*$" >nul
:: Capture du code de sortie de findstr
set findstr_exitcode=%errorlevel%
rem echo %findstr_exitcode%
:: Attention, le nombre peut être aussi -ve
echo %grade% | findstr /r "^[0-9]*$" >nul
:: Capture du code de sortie de findstr
set findstr_exitcode2=%errorlevel%
rem echo %findstr_exitcode2%

if %findstr_exitcode% neq 0 (
    if %findstr_exitcode2% neq 0 (
        echo error: the grade is a value between 0 and 20 but you typed "%grade%"
        exit /b 1
    ) else (
        rem echo Test passed 2 "Nombre -ve"
        echo error: the grade is a value between 0 and 20 but you typed "%grade%"
        exit /b 1
    )
) else (
    rem echo Test passed "Nombre +ve"
    :: Test de l'intervalle [0,20]
    if %grade% geq 0 if %grade% leq 20 (
        rem echo Test 3 passed
        :: Intervalle [0,9]
        if %grade% geq 0 if %grade% leq 9 (
            echo failure
            exit /b 0
        ) else (
            echo success
            exit /b 0
        )
    ) else (
        echo error: the grade is a value between 0 and 20 but you typed "%grade%"
        exit /b 1
    )
)
)
```

Figure 11



```
C:\Users\user\Documents\tp8-jean-paul-melisse>04-grade.bat
grade? 4
failure

C:\Users\user\Documents\tp8-jean-paul-melisse>echo %errorlevel%
0

C:\Users\user\Documents\tp8-jean-paul-melisse>04-grade.bat
grade? 15
success

C:\Users\user\Documents\tp8-jean-paul-melisse>echo %errorlevel%
0

C:\Users\user\Documents\tp8-jean-paul-melisse>04-grade.bat
grade? 25
error: the grade is a value between 0 and 20 but you typed "25"

C:\Users\user\Documents\tp8-jean-paul-melisse>echo %errorlevel%
1

C:\Users\user\Documents\tp8-jean-paul-melisse>04-grade.bat
grade? aas
error: the grade is a value between 0 and 20 but you typed "aas"

C:\Users\user\Documents\tp8-jean-paul-melisse>
```

Figure 12

## 10 Les expression arithmétiques en Batch

Les expressions arithmétiques permettent d'effectuer des calculs (additions, soustractions, . . . ). On crée un script 05-calculator.bat qui lit au clavier une chaîne de caractères représentant un opérateur, lit un entier représentant une première opérande, lit un entier représentant une seconde opérande et applique l'opérateur sur les opérandes. Si l'opérateur n'est pas supporté, le script doit écrire un message d'erreur et retourner 1 au lieu de 0. Si une opérande ne peut pas être convertie vers un entier, le script doit écrire un message d'erreur et retourner 1 au lieu de 0. Pour cela, on ouvre l'application Notepad. Dans ce nouveau fichier, on écrit les lignes suivantes :

- 1 @echo off
- 2 setlocal enabledelayedexpansion
- 3
- 4 echo addition
- 5 echo subtraction



```

6 echo multiplication
7 echo division
8 echo remainder
9 echo exponentiation
10
11 set /p operator="operator? "
12
13 set /p first_operand="first operand? "
14 echo %first_operand%| findstr /r "[0-9]*$" >nul
15 set findstr_exitcode=%errorlevel%
16 :: Nombre -ve
17 echo %first_operand%| findstr /r "^[0-9]*$" >nul
18 set findstr_exitcode2=%errorlevel%
19
20 if %findstr_exitcode% neq 0 (
21     if %findstr_exitcode2% neq 0 (
22         echo error: unable to convert 1st operand to an integer
23         exit /b 1
24     )
25 )
26
27 set /p second_operand="second operand? "
28 echo %second_operand%| findstr /r "[0-9]*$" >nul
29 set findstr_exitcode3=%errorlevel%
30 :: Nombre -ve
31 echo %second_operand%| findstr /r "^[0-9]*$" >nul
32 set findstr_exitcode4=%errorlevel%
33
34 if %findstr_exitcode3% neq 0 (

```

```

35     if %findstr_exitcode4% neq 0 (
36         echo error: unable to convert 2nd operand to an integer
37         exit /b 1
38     )
39 )
40 :: Opérations
41 if %operator%==addition (
42     set /a result=!first_operand! + !second_operand!
43     echo result: !result!
44     exit /b 0
45 )
46
47 if %operator%==subtraction (
48     set /a result=!first_operand! - !second_operand!
49     echo result: !result!
50     exit /b 0
51 )
52
53 if %operator%==multiplication (
54     set /a result=!first_operand! * !second_operand!
55     echo result: !result!
56     exit /b 0
57 )
58
59 if %operator%==division (
60     if !second_operand!==0 (
61         echo error: Division by zero
62         exit /b 1
63     ) else (

```

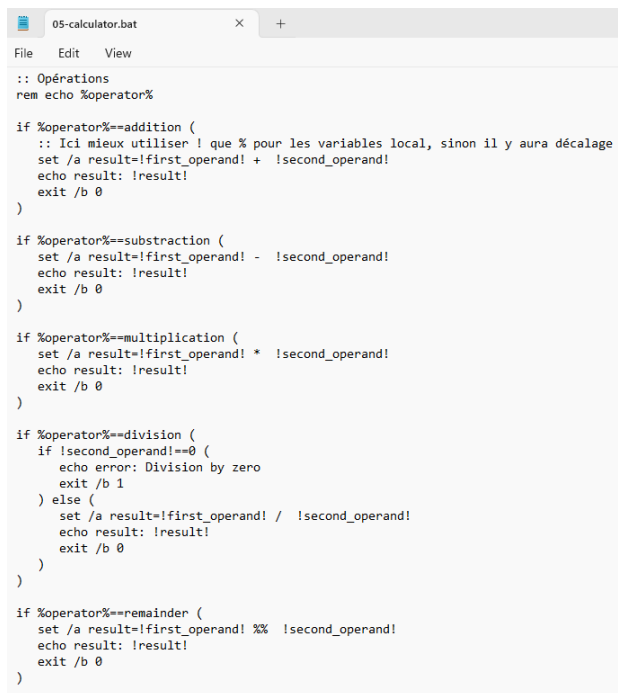
```

64         set /a result=!first_operand! + !second_operand!
65         echo result: !result!
66         exit /b 0
67     )
68 )
69
70 if %operator%==remainder (
71     set /a result=!first_operand! %% !second_operand!
72     echo result: !result!
73     exit /b 0
74 )
75
76 if %operator%==exponentiation (
77     if !second_operand! leq -1 (
78         echo error: 2nd operand should be +ve in exponentiation
79         exit /b 1
80     ) else (
81         set /a result=1
82         for /l %%i in (1,1,!second_operand!) do (
83             set /a result=!result!*!first_operand!
84         )
85         echo result: !result!
86         exit /b 0
87     )
88 )
89
90 :: Pas dans la liste des opérateurs (opérateur inconnu)
91 echo error: the operator is addition, subtraction, multiplication, division, remainder or
92 echo exponentiation but you typed "%operator%"

```

93 exit /b 1

On sauvegarde en utilisant "Ctrl+S" et on enregistre le fichier sous le nom "05-calculator.bat" dans le dossier tp8. On peut voir une partie de l'étape précédente sur la Figure 13. On peut vérifier l'exécution du script en ouvrant l'invite de commande. On appuie sur "Win + R", ce qui lance le programme "RUN". Dans le programme RUN, on tape "cmd" et on clique sur "OK". Sur l'invite de commande, on change de dossier en tapant la commande "cd Documents\tp8-jean-paul-melisse". Puis, on tape la commande : "05-calculator.bat" (voir la Figure 14).



```
05-calculator.bat
File Edit View
:: Opérations
rem echo %operator%

if %operator%==addition (
    :: Ici mieux utiliser ! que % pour les variables local, sinon il y aura décalage
    set /a result=!first_operand! + !second_operand!
    echo result: !result!
    exit /b 0
)

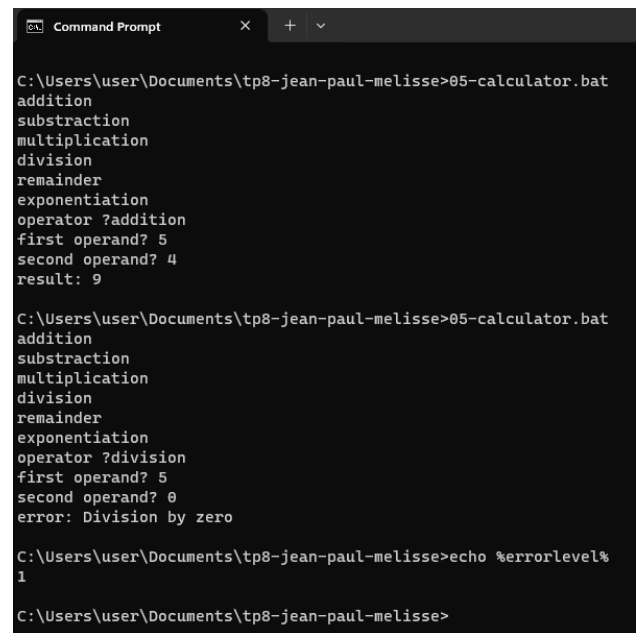
if %operator%==substraction (
    set /a result=!first_operand! - !second_operand!
    echo result: !result!
    exit /b 0
)

if %operator%==multiplication (
    set /a result=!first_operand! * !second_operand!
    echo result: !result!
    exit /b 0
)

if %operator%==division (
    if !second_operand!==0 (
        echo error: Division by zero
        exit /b 1
    ) else (
        set /a result=!first_operand! / !second_operand!
        echo result: !result!
        exit /b 0
    )
)

if %operator%==remainder (
    set /a result=!first_operand! %% !second_operand!
    echo result: !result!
    exit /b 0
)
```

Figure 13



```
Command Prompt
C:\Users\user\Documents\tp8-jean-paul-melisse>05-calculator.bat
addition
substraction
multiplication
division
remainder
exponentiation
operator ?addition
first operand? 5
second operand? 4
result: 9

C:\Users\user\Documents\tp8-jean-paul-melisse>05-calculator.bat
addition
substraction
multiplication
division
remainder
exponentiation
operator ?division
first operand? 5
second operand? 0
error: Division by zero

C:\Users\user\Documents\tp8-jean-paul-melisse>echo %errorlevel%
1

C:\Users\user\Documents\tp8-jean-paul-melisse>
```

Figure 14

## 11 Les itérations en Batch

Bien que les instructions d'un script soient habituellement exécutées de façon séquentielle, il est possible d'en exécuter à l'intérieur d'une structure itérative (boucle). Contrairement à d'autres langages, le Batch ne propose pas de structure tant que (while). Afin de simuler un tant que, nous utilisons une ligne étiquetée (labeled line), une structure conditionnelle et la commande goto. Ce script doit retourner 0.

- Syntaxe :

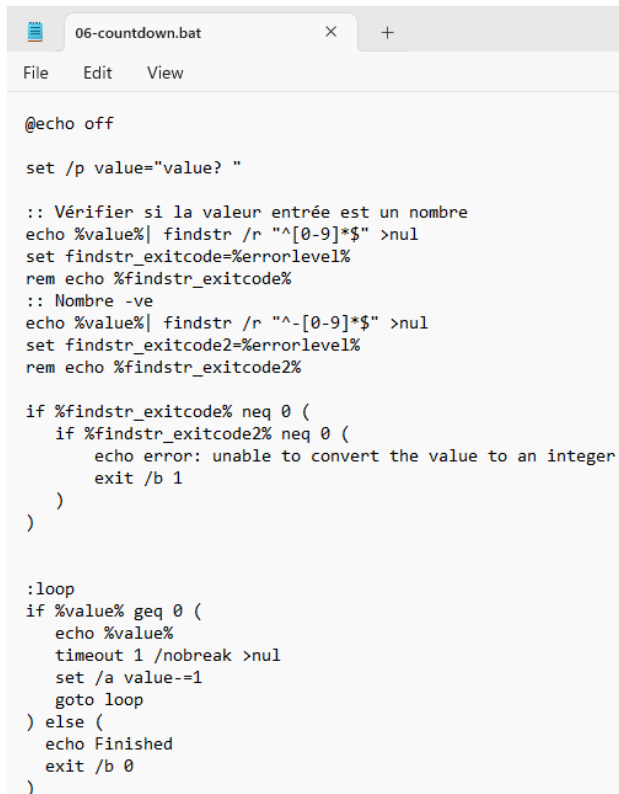
```
1 :loop
2 if CONDITION-1 (
3     STATEMENT-1
4     goto loop
5 )
```

On va donc créer un script 06-countdown.bat qui lit au clavier un entier représentant le début d'un compte à rebours, écrit cette valeur à l'écran, lui retire un, attend une seconde et recommence jusqu'à atteindre

zéro. Si le début du compte à rebours ne peut pas être converti vers un entier, le script doit écrire un message d'erreur et retourner 1 au lieu de 0. On ouvre l'application Notepad. Dans ce nouveau fichier, on écrit les lignes suivantes :

```
1 @echo off
2 set /p value="value? "
3
4 :: Vérifier si la valeur entrée est un nombre
5 echo %value%| findstr /r [0-9]*$
6 set findstr_exitcode=%errorlevel%
7 :: Nombre -ve
8 echo %value%| findstr /r "^[0-9]*$" >nul
9 set findstr_exitcode2=%errorlevel%
10
11 if %findstr_exitcode% neq 0 (
12     if %findstr_exitcode2% neq 0 (
13         echo error: unable to convert the value to an integer
14         exit /b 1
15     )
16 )
17
18 :loop
19 if %value% geq 0 (
20     echo %value%
21     timeout 1 /nobreak >nul
22     set /a value -=1
23     goto loop
24 ) else (
25     echo Finished
26     exit /b 0
```

On sauvegarde en utilisant "Ctrl+S" et on enregistre le fichier sous le nom "06-countdown.bat" dans le dossier tp8. On peut voir l'étape précédente sur la Figure 15. On peut vérifier l'exécution du script en ouvrant l'invite de commande. On appuie sur "Win + R", ce qui lance le programme "RUN". Dans le programme RUN, on tape "cmd" et on clique sur "OK". Sur l'invite de commande, on change de dossier en tapant la commande "cd Documents\tp8-jean-paul-melisse". Puis, on tape la commande : "06-countdown.bat" (voir la Figure 16).



```
@echo off

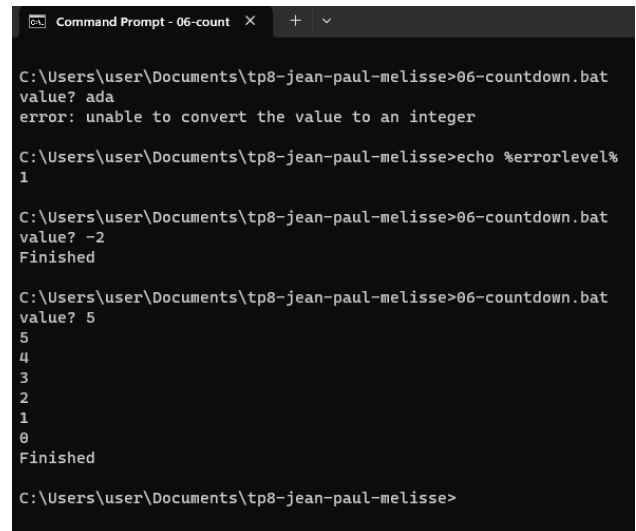
set /p value="value? "

:: Vérifier si la valeur entrée est un nombre
echo %value%| findstr /r "^[0-9]*$" >nul
set findstr_exitcode=%errorlevel%
rem echo %findstr_exitcode%
:: Nombre -ve
echo %value%| findstr /r "^[0-9]*$" >nul
set findstr_exitcode2=%errorlevel%
rem echo %findstr_exitcode2%

if %findstr_exitcode% neq 0 (
    if %findstr_exitcode2% neq 0 (
        echo error: unable to convert the value to an integer
        exit /b 1
    )
)

:loop
if %value% geq 0 (
    echo %value%
    timeout 1 /nobreak >nul
    set /a value-=1
    goto loop
) else (
    echo Finished
    exit /b 0
)
```

Figure 15



```
C:\Users\user\Documents\tp8-jean-paul-melisse>06-countdown.bat
value? ada
error: unable to convert the value to an integer

C:\Users\user\Documents\tp8-jean-paul-melisse>echo %errorlevel%
1

C:\Users\user\Documents\tp8-jean-paul-melisse>06-countdown.bat
value? -2
Finished

C:\Users\user\Documents\tp8-jean-paul-melisse>06-countdown.bat
value? 5
5
4
3
2
1
0
Finished

C:\Users\user\Documents\tp8-jean-paul-melisse>
```

Figure 16

## 12 Les fonctions en Batch

Les fonctions ont un nom, des paramètres et une valeur de retour. Elles permettent de décomposer le code en plusieurs calculs indépendants. Une fonction peut être définie une fois et appelée autant de fois que nécessaire. Contrairement à d'autres langages, le Batch ne propose pas les fonctions. Afin de simuler une fonction, nous utilisons une ligne étiquetée et la commande call.

- Syntaxe pour définir une fonction :

```
1 :NAME
2 STATEMENT-1
3 STATEMENT-2
```

- Syntaxe pour appeler une fonction :

```
1 call :NAME PARAMETER-VALUE-1
```

Les paramètres peuvent être obtenus grâce à \$1 pour le premier paramètre et ainsi de suite pour les autres paramètres. La valeur de retour peut être obtenue grâce à l'affectation d'une valeur à une variable. On va créer un script "07-even.bat" qui définit la fonction "even" à un paramètre "NUMBER" et qui retourne 0 si "NUMBER" est pair, sinon 2. Ce script lit au clavier un entier, appelle la fonction "even" en lui passant en paramètre l'entier précédemment lu au clavier et écrit à l'écran le résultat obtenu. Si le nombre ne peut pas être converti vers un entier, le script doit écrire un message d'erreur et retourner 1 au lieu de 0. Pour cela, on ouvre l'application Notepad. Dans ce nouveau fichier, on écrit les lignes suivantes :

```

1 @echo off
2 set /p number="number? "
3
4 :: Vérifier si la valeur entrée est un nombre
5 echo %number%| findstr /r [0-9]*$
6 set findstr_exitcode=%errorlevel%
7 :: Nombre -ve
8 echo %number%| findstr /r "^[0-9]*$" >nul
9 set findstr_exitcode2=%errorlevel%
10
11 if %findstr_exitcode% neq 0 (
12     if %findstr_exitcode2% neq 0 (
13         echo error: unable to convert the number to an integer
14         exit /b 1
15     )
16 )
17
18 call :even %number%
19 echo the number "%number%" is %result%
20 exit /b %exitcode%
21
22 rem #####
23 :: fonction even
24 :even

```

```

25 set num=%1

26 set /a remainder=!num! %% 2

27

28 if !remainder! == 0 (

29     set result=even

30     set exitcode=0

31 ) else (

32     set result=odd

33     set exitcode=2

34 )

35 goto :eof

```

On sauvegarde en utilisant "Ctrl+S" et on enregistre le fichier sous le nom "07-even.bat" dans le dossier tp8. On peut voir l'étape précédente sur la Figure 17. On peut vérifier l'exécution du script en ouvrant l'invite de commande. On appuie sur "Win + R", ce qui lance le programme "RUN". Dans le programme RUN, on tape "cmd" et on clique sur "OK". Sur l'invite de commande, on change de dossier en tapant la commande "cd Documents\tp8-jean-paul-melisse". Puis, on tape la commande : "07-even.bat" (voir la Figure 18).



```

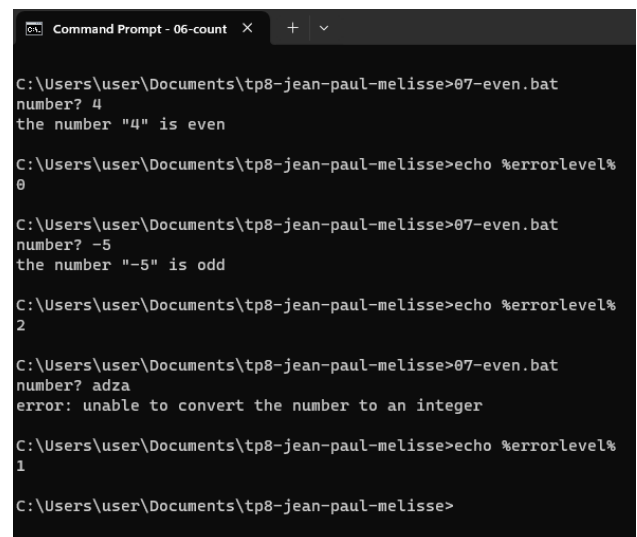
@echo off
setlocal enabledelayedexpansion
set /p number="number? "
:: Vérifier si l'entrée est un nombre
echo %number% | findstr /r "[0-9]*$" >nul
set findstr_exitcode=%errorlevel%
rem echo %findstr_exitcode%
:: Nombre -ve
echo %number% | findstr /r "[^-][0-9]*$" >nul
set findstr_exitcode2=%errorlevel%
rem echo %findstr_exitcode2%

if %findstr_exitcode% neq 0 (
    if %findstr_exitcode2% neq 0 (
        echo error: unable to convert the number to an integer
        exit /b 1
    )
)
call :even %number%
echo the number "%number%" is %result%
exit /b %exitcode%
rem #####
rem fonction even
:even
set num=%1
set /a remainder=!num! %% 2

if !remainder! == 0 (
    set result=even
    set exitcode=0
) else (
    set result=odd
    set exitcode=2
)
goto :eof

```

Figure 17



```

C:\Users\user\Documents\tp8-jean-paul-melisse>07-even.bat
number? 4
the number "4" is even

C:\Users\user\Documents\tp8-jean-paul-melisse>echo %errorlevel%
0

C:\Users\user\Documents\tp8-jean-paul-melisse>07-even.bat
number? -5
the number "-5" is odd

C:\Users\user\Documents\tp8-jean-paul-melisse>echo %errorlevel%
2

C:\Users\user\Documents\tp8-jean-paul-melisse>07-even.bat
number? adza
error: unable to convert the number to an integer

C:\Users\user\Documents\tp8-jean-paul-melisse>echo %errorlevel%
1

C:\Users\user\Documents\tp8-jean-paul-melisse>

```

Figure 18

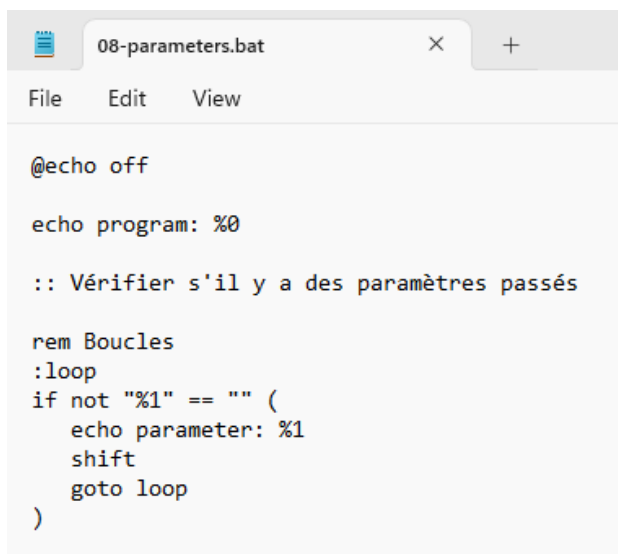


## 13 Les paramètres en Batch

Des paramètres peuvent être passés aux scripts en ligne de commande de la même manière que des paramètres peuvent être passés aux fonctions. On crée un script 08-parameters.bat qui écrit à l'écran le nom du script (paramètre zéro) et les paramètres passés. On ouvre l'application Notepad. Dans ce nouveau fichier, on écrit les lignes suivantes :

```
1 @echo off
2 echo program: %0
3
4 :: Vérifier s'il y a des paramètres passés
5 :loop
6 if not "%1" == "" (
7     echo parameter: %1
8     shift
9     goto loop
10 )
```

On sauvegarde en utilisant "Ctrl+S" et on enregistre le fichier sous le nom "08-parameters.bat" dans le dossier tp8. On peut voir l'étape précédente sur la Figure 19. On peut vérifier l'exécution du script en ouvrant l'invite de commande. On appuie sur "Win + R", ce qui lance le programme "RUN". Dans le programme RUN, on tape "cmd" et on clique sur "OK". Sur l'invite de commande, on change de dossier en tapant la commande "cd Documents\tp8-jean-paul-melisse". Puis, on tape la commande : "08-parameters.bat" (voir la Figure 20).

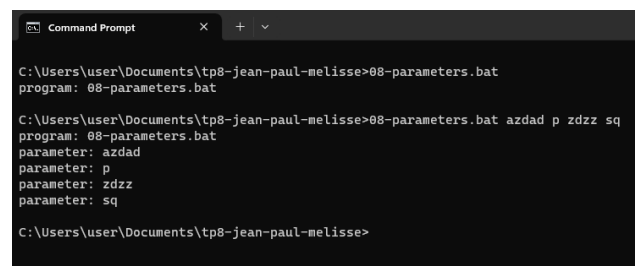
A screenshot of a Notepad window titled "08-parameters.bat". The window has a menu bar with "File", "Edit", and "View". The text inside the Notepad is as follows:

```
@echo off
echo program: %0

:: Vérifier s'il y a des paramètres passés

rem Boucles
:loop
if not "%1" == "" (
    echo parameter: %1
    shift
    goto loop
)
```

Figure 19

A screenshot of a Windows Command Prompt window titled "Command Prompt". It shows the execution of the batch file "08-parameters.bat" from the directory "C:\Users\user\Documents\tp8-jean-paul-melisse". The output is as follows:

```
C:\Users\user\Documents\tp8-jean-paul-melisse>08-parameters.bat
program: 08-parameters.bat

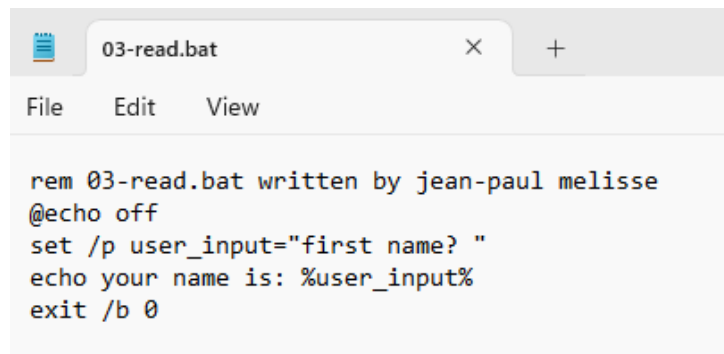
C:\Users\user\Documents\tp8-jean-paul-melisse>08-parameters.bat azdad p zdzz sq
program: 08-parameters.bat
parameter: azdad
parameter: p
parameter: zdzz
parameter: sq

C:\Users\user\Documents\tp8-jean-paul-melisse>
```

Figure 20

## 14 Les commentaires en Batch

Pour commenter un code, il suffit de précéder le texte par `rem`, exemple : `rem this is my comment`. On va ajouter un commentaire à la première ligne de tous les scripts qu'on a créés afin d'y indiquer "`rem <fichier> written by jean-paul melisse`" (avec `<fichier>` à remplacer par le nom du fichier). Seuls les caractères suivants sont autorisés pour ce commentaire : 0-9 (numériques), a-z (minuscules), - (tiret) et .(point). On peut voir un exemple sur la Figure 21.

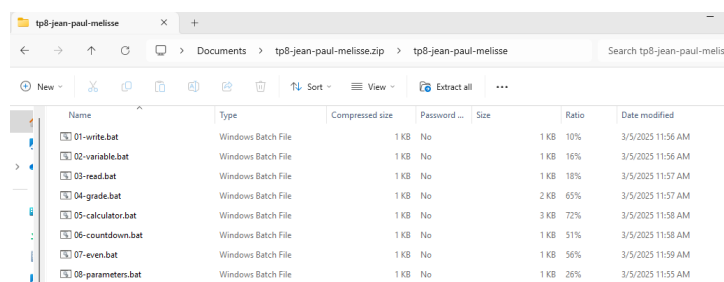


```
rem 03-read.bat written by jean-paul melisse
@echo off
set /p user_input="first name? "
echo your name is: %user_input%
exit /b 0
```

Figure 21

## 15 Création de l'archive et son exportation

Pour cette partie de ce TP, on va archiver le dossier "`tp8-jean-paul-melisse`" et l'exporter sur la machine physique. Pour commencer, on revient sur le dossier parent (`C:\Users\User\Documents`) avec Windows Explorer. On clique-droit sur le dossier "`tp8-jean-paul-melisse`" et on clique sur "`Compress to..`", puis sur "`ZIP File`". Cela va créer notre archive "`tp8-jean-paul-melisse.zip`". On peut vérifier le contenu de l'archive en l'ouvrant avec Windows Explorer (Voir la Figure 22).



Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
01-write.bat	Windows Batch File	1 KB	No	1 KB	10%	3/5/2025 11:56 AM
02-variable.bat	Windows Batch File	1 KB	No	1 KB	16%	3/5/2025 11:56 AM
03-read.bat	Windows Batch File	1 KB	No	1 KB	18%	3/5/2025 11:57 AM
04-grade.bat	Windows Batch File	1 KB	No	2 KB	65%	3/5/2025 11:57 AM
05-calculator.bat	Windows Batch File	1 KB	No	3 KB	72%	3/5/2025 11:58 AM
06-countdown.bat	Windows Batch File	1 KB	No	1 KB	51%	3/5/2025 11:58 AM
07-even.bat	Windows Batch File	1 KB	No	1 KB	56%	3/5/2025 11:59 AM
08-parameters.bat	Windows Batch File	1 KB	No	1 KB	26%	3/5/2025 11:55 AM

Figure 22

La suite est de créer un dossier partagé. On retourne sur la machine physique et on crée le dossier "`share`" dans le chemin : "`C:\Users\Jean-Paul\Desktop\tp8\`". Puis, on va sur VirtualBox et on clique sur "`tp8-windows`" et "`Configuration`". Puis on clique sur "`Shared Folders`" et sur "`Ajouter un nouveau dossier partagé`". On met dans le chemin du dossier : "`C:\Users\Jean-Paul\Desktop\tp8\share`". On met le nom du dossier : "`share`" et on clique sur "`Ok`". On peut voir sur la Figure 23 que le dossier "`share`" a bien été partagé avec la machine virtuelle "`tp8-windows`".

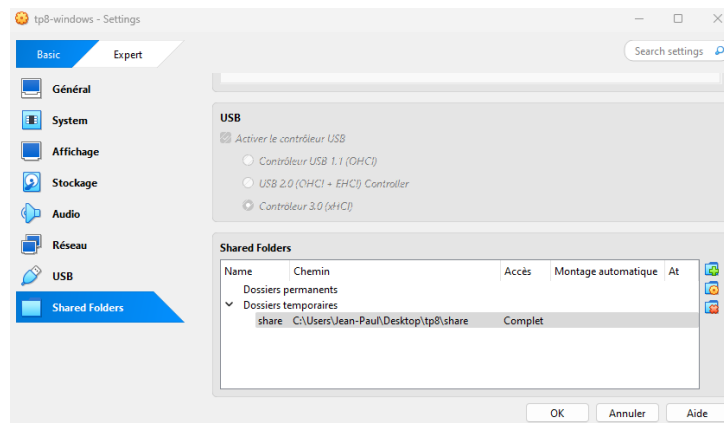


Figure 23

On retourne sur la machine "tp8-windows". On ouvre Windows Explorer et on clique sur l'onglet "Network". On clique sur "Turn on Network discovery" pour voir les ordinateurs connectés au réseau. On passe le réseau actuel en réseau privé et on voit apparaître l'hôte "VBOXSVR". On entre dans ce dossier, puis dans le dossier share. Enfin, on copie l'archive "tp8-jean-paul-melisse.zip" dans ce dossier en faisant un "glisser/déposer" ou un "copier/coller". On retourne sur la machine physique et on regarde le dossier "share". On voit, comme dans la Figure 24, que l'archive est bien sur la machine physique. On peut l'exporter pour une utilisation ultérieure.

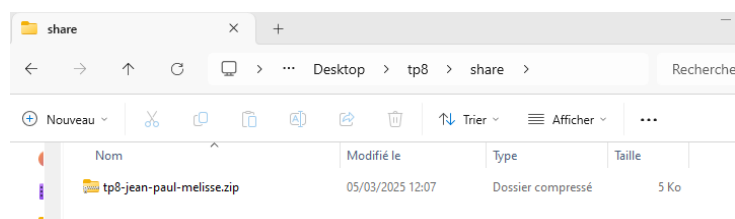


Figure 24