

1. Készítsünk scriptet, amelyik a munkakönyvtár bejegyzései közül kiírja a fájlokat!

```
Write-Output "Kiírom az aktuális könyvtár fájljait"  
Get-ChildItem -File
```

2. Készítsünk scriptet, amelyik meghatározza egy paraméterként átadott fájl sorainak a számát!

```
Get-Content $args[0] | Measure-Object -Line | Format-List Lines
```

vagy

```
$sorszam = Get-Content $args[0] | Measure-Object -Line  
Write-Host Az $args[0] fájl sorainak száma: $sorszam.Lines
```

3. Készítsünk scriptet, amelyik az első és második paramétert összeadja illetve kiírja a szorzatukat is!

```
Write-Host Kiíratom az első és második paraméter összegét és szorzatát:  
$osszeg = $args[0]+$args[1]  
$szorzat = $args[0]*$args[1]  
Write-Host $args[0] és $args[1] összege: $osszeg és szorzata: $szorzat
```

4. Készítsünk scriptet, amelyik egy fájl soraiban lévő szavak közül kiírja a másodikikat! A szavak között legyen szóköz. A fájlnevet paraméterként kapja.

```
write-host A paraméterként kapott fájl sorainak második szavai:  
foreach ($i in Get-Content $args[0])  
{  
    write-host $i.Split(' ')[1]  
}
```

5. Készítsünk scriptet, amelyik egy fájl sorainak 2-4 karaktereit írja ki! A fájlnevet paraméterként kapja.

```
write-host A paraméterként kapott fájl sorainak 2-4. karaktere:  
foreach ($i in Get-Content $args[0])  
{  
    write-host $i.Substring(1,3)  
}
```

6. Készítsünk scriptet, amelyik egy fájl sorainak megadott sorszámok közötti karaktereit írja ki! A fájlnevet paraméterként kapja, a sorszámokat futás közben olvassa be!

```
$a = Read-Host -Prompt "Kérem a kisebb sorszámot"
$b = Read-Host -Prompt "Kérem a nagyobb sorszámot"
write-host A paraméterként kapott fájl sorainak $a-$b. karaktere:
foreach ($i in Get-Content $args[0])
{
    write-host $i.Substring($a-1,$b-$a+1)
}
```

7. Készítsünk scriptet, amelyik egy fájl sorainak megadott sorszámok közötti karaktereit írja ki! A fájlnevet paraméterként kapja, a sorszámokat futás közben olvassa be! Ellenőrizze, hogy a számok sorrendje nem fordított-e.

```
$a = Read-Host -Prompt "Kérem a kisebb sorszámot"
$b = Read-Host -Prompt "Kérem a nagyobb sorszámot"
if ($a -le $b)
{
    write-host A paraméterként kapott fájl sorainak $a-$b. karaktere:
    foreach ($i in Get-Content $args[0])
    {
        write-host $i.Substring($a-1,$b-$a+1)
    }
}
else
{
    Write-Host Hibás sorszámmegadás!
}
```

8. Készítsük el a 7. feladatban leírt scriptet úgy, hogy ha a számok sorrendje fordított, cserélje meg azokat. Ne jelenítsen meg hibaüzeneteket azoknál a soroknál, melyekben nincs megfelelő számúkarakter!

```
$ErrorActionPreference="SilentlyContinue"
$a = Read-Host -Prompt "Kérem a kisebb sorszámot"
$b = Read-Host -Prompt "Kérem a nagyobb sorszámot"
if ($a -gt $b)
{
    $c = $a
    $a = $b
    $b = $c
}
write-host A paraméterként kapott fájl sorainak $a-$b. karaktere:
foreach ($i in Get-Content $args[0])
{
    write-host $i.Substring($a-1,$b-$a+1)
}
```

9. Készítsünk scriptet, mely egy menüt valósít meg. A menüben 3 menüpont legyen. Az első kettő 2 másodpercre kiírja a képernyőre, hogy mit választott a felhasználó. A 3. legyen a kilépés. Minden egyéb választás esetén figyelmeztesse a felhasználót, hogy rosszat választott.

```
do {
    Clear-Host
    Write-Host Kérem válasszon!
    Write-Host 1. Első menüpont
    Write-Host 2. Második menüpont
    Write-Host 3. Kilépés
    $választás = Read-Host
    switch ($választás)
    {
        1 {Write-Host Ön az első menüpontot választotta !}
        2 {Write-Host Ön a második menüpontot választotta!}
        3 {Write-Host Ön a kilépést választotta!}
        default {Write-Host Hibás választás!}
    }
    Start-Sleep -Seconds 2
} while ($választás -ne 3)
```

10. Készítsünk scriptet, amelyik kiszámolja a paraméterben megadott szám faktoriálisát! Ellenőrizze, hogy 1 paramétert kapott-e. Ha nem, akkor jelezze a hibát és a helyes szintaxist.

```
if ($args.Length -ne 1 )
{
    Write-Host "Pontosan egy paraméter kell! A parancs helyes
    használata: $($MyInvocation.MyCommand.Name) <szám>"
}
else
{
    $fakt=1
    for ($i=1; $i -le $args[0]; $i++)
    {
        $fakt=$fakt*$i
    }
    Write-Host "A(z) $($args[0]) faktoriálisa: $fakt"
    Write-Host "A(z)"$args[0]"faktoriálisa: $fakt"
}
```

11. Készítsünk scriptet, amelyik kiszámolja a paraméterben megadott szám faktoriálisát! Kötelező paraméterként kérjük be a számot. Kényszerítsük a megfelelő típusú érték beírását!

```
param
(
    [Parameter(Mandatory=$true)][int]$n
)
$fakt=1
for ($i=1; $i -le $n; $i++)
{
    $fakt=$fakt*$i
}
Write-Host "A(z) $n faktoriálisa: $fakt"
```

12. Készítsünk scriptet, amelyik kiszámolja az összes paraméter összegét! Készítsük el a megoldást for és foreach használatával is!

```
$sum=0
for ($i=0; $i -lt $args.Length; ++$i)
{
    $sum=$sum+$args[$i]
}
Write-Output "A paraméterek összege: $sum"
```

vagy

```
$sum=0
foreach ($i in $args)
{
    $sum=$sum+$i
}
Write-Output "A paraméterek összege: $sum"
```

13. Módosítsuk az előző scriptet úgy, hogy szűrőként is használhassuk!

```
$sum=0
if (@($input).Count -ne 0)
{
    $input.Reset()
    foreach($i in $input)
    {
        $szum+=$i
    }
}
else
{
    foreach ($i in $args)
    {
        $szum+=$i
    }
}
Write-Host $szum
```

14. Készítsünk el egy másodfokú egyenletet kiszámító scriptet, amelyik billentyűzetről olvassa be az adatokat!

```
Write-Host "Másodfokú egyenlet megoldó program"
[double]$a=read-host "Írd be az ""a"" együtthatót"
[double]$b=read-host "Írd be a ""b"" együtthatót"
[double]$c=read-host "Írd be a ""c"" együtthatót"
Write-Host "Az egyenlet: "$a"*x2+"$b"*x+"$c"=0"
$diszkriminans=[Math]::pow($b,2)-4*$a*$c
if ($diszkriminans -lt 0)
{
    Write-Host "Nincs megoldás"
}
elseif ($diszkriminans -eq 0)
{
    $x1=(-$b)/2*$a
    Write-Host "Egy megoldás van: "$x1
}
else
{
    $x1=((-$b)+[Math]::sqrt($diszkriminans))/(2*$a)
    $x2=((-$b)-[Math]::sqrt($diszkriminans))/(2*$a)
    Write-Host "Az egyenlet megoldásai: "$x1" és "$x2
}
```

15. Készítsünk scriptet, amelyik egy fájl páros és páratlan sorait külön-külön fájlokba másolja (PárosSorok.txt és PáratlanSorok.txt)! Ellenőrizzük, hogy létezik-e a fájl!

```
if (Test-Path $args[0])
{
    if (Test-Path PárosSorok.txt)
    {
        Remove-Item PárosSorok.txt
    }
    if (Test-Path PáratlanSorok.txt)
    {
        Remove-Item PáratlanSorok.txt
    }
    New-Item -Name PárosSorok.txt -ItemType File
    New-Item -Name PáratlanSorok.txt -ItemType File
    [bool]$Páros=$true
    foreach ($i in Get-Content $args[0])
    {
        if ($Páros)
        {
            Add-Content PárosSorok.txt -Value $i
        }
        else
        {
            Add-Content PáratlanSorok.txt -Value $i
        }
        $Páros= -not $Páros
    }
}
else
{
    Write-Host "Nincs ilyen fájl!"
}
```

16. Készítsünk scriptet, amelyik egy természetes számról eldönti, hogy prím-e!

```
[int]$szám=read-host -Prompt "Írj be egy természetes számot"
Write-Output $szám
if ($szám -lt 0)
{
    Write-Output "Nem természetes számot írtál be!"
}
elseif ($szám -eq 0)
{
    Write-Output "A szám nem prím."
}
elseif ($szám -eq 1)
{
    Write-Output "A szám sem nem prím, sem nem összetett."
}
elseif ($szám -eq 2)
{
    Write-Output "A szám prím."
}
else
{
    $i=2
    while ($i -lt $szám -and $szám % $i -ne 0)
    {
        $i++
    }
    if ($i -lt $szám)
    {
        Write-Output "A szám nem prím."
    }
    else
    {
        Write-Output "A szám prím."
    }
}
```

17. Készítsünk scriptet, amelyik meghatározza két egész szám legnagyobb közös osztóját!

```
[int]$a=read-host -Prompt "Kérem az első számot"
[int]$b=read-host -Prompt "Kérem a második számot"
if ($a -lt 0)
{
    $a*=-1
}
if ($b -lt 0)
{
    $b*=-1
}
$i=$a
if ($b -lt $a)
{
    $i=$b
}
while (($a % $i -ne 0) -or ($b % $i -ne 0))
{
    --$i
}
Write-Host "Legnagyobb közös osztójuk: "$i
```