[1. Використовуючи засоби PoweShell розробити рекурсивну функцію для обчислення факторіала (n!) натурального числа. За її допомогою обчислити величину](#_Toc27433189)

[2. Використовуючи засоби PowerShell, розробити рекурсивну функцію для обчислення факторіала (n! ) натурального числа. За її допомогою обчислити величину : (24!) - (13) + (20!).](#_Toc27433190)

[3. Використовуючи засоби PowerShell, розробити рекурсивну функцію для обчислення факторіала (n!) натурального числа. За її допомогою обчислити величину: (n! + 34)! + (n - 3)! - (2n!).](#_Toc27433191)

[4. Використовуючи засоби PowerShell, розробити функцію для обчислення довжини відрізку на площині, за її допомогою обчислити периметр трикутника, заданого координатами своїх вершин](#_Toc27433192)

[5. Використовуючи засоби PowerShell, розробити функцію для обчислення довжини відрізка на площині. За її допомогою обчислити периметр чотирикутника, заданого координатами своїх вершин.](#_Toc27433193)

[6. Використовуючи засоби PowerShell, розробити функцію для обчислення довжини відрізка на площині. За її допомогою обчислити периметр п’ятикутника, заданого координатами своїх вершин.](#_Toc27433194)

[7.Використовуючи засоби PowerShell, розробити функцію для обчислення довжини відрізка у тривимірному просторі. За її допомогою обчислити периметр трикутної піраміди, заданої координатами своїх вершин.](#_Toc27433195)

[8.Використовуючи засоби PowerShell, розробити функцію для обчислення довжини відрізка у тривимірному просторі. За її допомогою обчислити периметр куба, заданої координатами своїх вершин.](#_Toc27433196)

[9. Використовуючи засоби PowerShell, розробити функцію для обчислення довжини відрізку у тривимірному просторі, за її допомогою обчислити периметр паралеепіпеда, заданого координатами своїх вершин](#_Toc27433197)

[10.Використовуючи засоби PowerShell, розробити функцію для обчислення довжини відрізка у тривимірному просторі. За її допомогою обчислити об’єм куба, заданої координатами своїх вершин.](#_Toc27433198)

[11.Використовуючи засоби PowerShell, розробити функцію для обчислення довжини відрізка у тривимірному просторі. За її допомогою обчислити об’єм прямокутного паралелепіпеда, заданої координатами своїх вершин.](#_Toc27433199)

[12. Використовуючи засоби PowerShell, розробити функцію для визначення кількості цифр у файлі.](#_Toc27433200)

[13. Використовуючи засоби PowerShell, розробити функцію для визначення кількості англійських літер у заданому файлі](#_Toc27433201)

[14. Використовуючи засоби PowerShell, розробити функцію для визначення кількості розділових знаків у файлі.](#_Toc27433202)

[15. Використовуючи засоби PowerShell, розробити функцію для визначення кількості українських літер у заданому файлі](#_Toc27433203)

# 1. Використовуючи засоби PoweShell розробити рекурсивну функцію для обчислення факторіала (n!) натурального числа. За її допомогою обчислити величину

function Get-Factorial ($n) {

if ($n -eq 0) {

return 1

}

$fact = 1

1..$n | ForEach { $fact \*= $\_ }

return $fact

}

$n = 10

$m = 2

foreach ($i in 0..$n) {

echo ("{0}! = {1}" -f $i,(Get-Factorial $i))

}

$C = ((Get-Factorial($n))/((Get-Factorial($m))\*(Get-Factorial($n-$m))))

echo $C

# 2. Використовуючи засоби PowerShell, розробити рекурсивну функцію для обчислення факторіала (n! ) натурального числа. За її допомогою обчислити величину : (24!) - (13) + (20!).

function fact($n){

if($n -eq 1){return 1;}

else{return $n \* (fact($n-1));}

}

$result = fact(24) - fact(13) + fact(20);

$result

# 3. Використовуючи засоби PowerShell, розробити рекурсивну функцію для обчислення факторіала (n!) натурального числа. За її допомогою обчислити величину: (n! + 34)! + (n - 3)! - (2n!).

[float]$n=read-host

function fact($n){

if($n -eq 1){return 1;}

else{return $n \* (fact($n-1));}

}

$result = fact($n + 34) + fact($n - 3) - (2 \*( fact($n)));

$result

# 4. Використовуючи засоби PowerShell, розробити функцію для обчислення довжини відрізку на площині, за її допомогою обчислити периметр трикутника, заданого координатами своїх вершин

function GetLength($x1, $y1, $x2, $y2)

{

return [Math]::Sqrt([Math]::Pow(($x1-$x2),2)+[Math]::Pow(($y1-$y2),2))

}

[int]$ax = Read-Host -Prompt 'Input A(x)'

[int]$ay = Read-Host -Prompt 'Input A(y)'

[int]$bx = Read-Host -Prompt 'Input B(x)'

[int]$by = Read-Host -Prompt 'Input B(y)'

[int]$cx = Read-Host -Prompt 'Input C(x)'

[int]$cy = Read-Host -Prompt 'Input C(y)'

$l1 = GetLength $ax $ay $bx $by

$l2 = GetLength $bx $by $cx $cy

$l3 = GetLength $cx $cy $ax $ay

$p = $l1 + $l2 + $l3

Write-Host "Perimeter = $p"

# 5. Використовуючи засоби PowerShell, розробити функцію для обчислення довжини відрізка на площині. За її допомогою обчислити периметр чотирикутника, заданого координатами своїх вершин.

Write-Host "Введiть xa:"  
[float]$xa=read-host  
Write-Host "Введiть ya:"  
[float]$ya=read-host  
Write-Host "Введiть xb:"  
[float]$xb=read-host  
Write-Host "Введiть yb:"  
[float]$yb=read-host  
Write-Host "Введiть xc:"  
[float]$xc=read-host  
Write-Host "Введiть yc:"  
[float]$yc=read-host  
Write-Host "Введiть xd:"  
[float]$xd=read-host  
Write-Host "Введiть yd:"  
[float]$yd=read-host  
  
function length($x1, $y1, $x2, $y2) {  
$len= [Math]::Sqrt([Math]::Pow(($x2-$x1),2)+[Math]::Pow(($y2-$y1),2))  
  
return $len  
}  
  
$AB = length $xa $ya $xb $yb  
$BC = length $xb $yb $xc $yc  
$CD = length $xc $yc $xd $yd  
$DA = length $xd $yd $xa $ya  
  
$Perim = $AB+$BC+$CD+$DA

Write-Host "Периметр = " $Perim

# 6. Використовуючи засоби PowerShell, розробити функцію для обчислення довжини відрізка на площині. За її допомогою обчислити периметр п’ятикутника, заданого координатами своїх вершин.

Write-Host "Введiть xa:"  
[float]$xa=read-host  
Write-Host "Введiть ya:"  
[float]$ya=read-host  
Write-Host "Введiть xb:"  
[float]$xb=read-host  
Write-Host "Введiть yb:"  
[float]$yb=read-host  
Write-Host "Введiть xc:"  
[float]$xc=read-host  
Write-Host "Введiть yc:"  
[float]$yc=read-host  
Write-Host "Введiть xd:"  
[float]$xd=read-host  
Write-Host "Введiть yd:"  
[float]$yd=read-host  
Write-Host "Введiть xe:"  
[float]$xe=read-host  
Write-Host "Введiть ye:"  
[float]$ye=read-host  
  
function length($x1, $y1, $x2, $y2) {  
$len= [Math]::Sqrt([Math]::Pow(($x2-$x1),2)+[Math]::Pow(($y2-$y1),2))  
  
return $len  
}  
  
$AB = length $xa $ya $xb $yb  
$BC = length $xb $yb $xc $yc  
$CD = length $xc $yc $xd $yd  
$DE = length $xd $yd $xe $ye  
$EA = length $xe $ye $xa $ya  
  
$Perim = $AB+$BC+$CD+$DE+$EA  
  
Write-Host "Периметр = " $Perim

# 7.Використовуючи засоби PowerShell, розробити функцію для обчислення довжини відрізка у тривимірному просторі. За її допомогою обчислити периметр трикутної піраміди, заданої координатами своїх вершин.

#ф-я для обч довжини відрізка у 3D

function segmentLenght($x1, $y1, $x2, $y2, $z1, $z2) {

$len= [Math]::Sqrt([Math]::Pow(($x2-$x1),2)+[Math]::Pow(($y2-$y1),2)+[Math]::Pow(($z2-$z1),2))

return $len

}

#координати вершин

Write-Host "Вкажіть вершину піраміди: "

$xA = Read-Host

$yA = Read-Host

$zA = Read-Host

Write-Host "Вкажіть вершину основи піраміди: "

$xB = Read-Host

$yB = Read-Host

$zB = Read-Host

Write-Host "Вкажіть вершину основи піраміди: "

$xC = Read-Host

$yC = Read-Host

$zC = Read-Host

Write-Host "Вкажіть вершину основи піраміди: "

$xD = Read-Host

$yD = Read-Host

$zD = Read-Host

#знаходимо довжину ребер

#бічне ребро

$AB = segmentLenght $xA $yA $zA $xB $yB $zB

$BD = segmentLenght $xB $yB $zB $xD $yD $zD

$DC = segmentLenght $xD $yD $zD $xC $yC $zC

#бічне ребро

$CA = segmentLenght $xC $yC $zC $xA $yA $zA

$BC = segmentLenght $xB $yB $zB $xC $yC $zC

#бічне ребро

$AD = segmentLenght $xA $yA $zA $xD $yD $zD

#перевірка на рівність бічних ребер

if($AB -eq $CA -and $CA -eq $AD)

{

$Perimeter = $AB + $BD + $DC + $CA + $BC + $AD

Write-Host $Perimeter

}

else

{

Write-Host "Invalid values"

}

# 8.Використовуючи засоби PowerShell, розробити функцію для обчислення довжини відрізка у тривимірному просторі. За її допомогою обчислити периметр куба, заданої координатами своїх вершин.

function GetLength($x1, $y1, $z1, $x2, $y2, $z2)

{

return [Math]::Sqrt([Math]::Pow(($x1-$x2),2)+[Math]::Pow(($y1-$y2),2)+[Math]::Pow(($z1-$z2),2))

}

[int]$ax = Read-Host -Prompt 'Input A(x)'

[int]$ay = Read-Host -Prompt 'Input A(y)'

[int]$az = Read-Host -Prompt 'Input A(z)'

[int]$bx = Read-Host -Prompt 'Input B(x)'

[int]$by = Read-Host -Prompt 'Input B(y)'

[int]$bz = Read-Host -Prompt 'Input B(z)'

[int]$cx = Read-Host -Prompt 'Input C(x)'

[int]$cy = Read-Host -Prompt 'Input C(y)'

[int]$cz = Read-Host -Prompt 'Input C(z)'

[int]$dx = Read-Host -Prompt 'Input D(x)'

[int]$dy = Read-Host -Prompt 'Input D(y)'

[int]$dz = Read-Host -Prompt 'Input D(z)'

[int]$ax1 = Read-Host -Prompt 'Input A1(x)'

[int]$ay1 = Read-Host -Prompt 'Input A1(y)'

[int]$az1 = Read-Host -Prompt 'Input A1(z)'

[int]$bx1 = Read-Host -Prompt 'Input B1(x)'

[int]$by1 = Read-Host -Prompt 'Input B1(y)'

[int]$bz1 = Read-Host -Prompt 'Input B1(z)'

[int]$cx1 = Read-Host -Prompt 'Input C1(x)'

[int]$cy1 = Read-Host -Prompt 'Input C1(y)'

[int]$cz1 = Read-Host -Prompt 'Input C1(z)'

[int]$dx1 = Read-Host -Prompt 'Input D1(x)'

[int]$dy1 = Read-Host -Prompt 'Input D1(y)'

[int]$dz1 = Read-Host -Prompt 'Input D1(z)'

$ab = GetLength $ax $ay $az $bx $by $dz

$bc = GetLength $bx $by $dz $cx $cy $cz

$cd = GetLength $cx $cy $cz $dx $dy $dz

$da = GetLength $dx $dy $dz $ax $ay $az

$a1b1 = GetLength $ax1 $ay1 $az1 $bx1 $by1 $dz1

$b1c1 = GetLength $bx1 $by1 $dz1 $cx1 $cy1 $cz1

$c1d1 = GetLength $cx1 $cy1 $cz1 $dx1 $dy1 $dz1

$d1a1 = GetLength $dx1 $dy1 $dz1 $ax1 $ay1 $az1

$aa1 = GetLength $ax $ay $az $ax1 $ay1 $az1

$bb1 = GetLength $bx $by $dz $b1x $by1 $bz1

$cc1 = GetLength $cx $cy $cz $cx1 $cy1 $cz1

$dd1 = GetLength $dx $dy $dz $dx1 $dy1 $dz1

if($ab -eq $bc –and $bc –eq $cd -and $cd –eq $da -and $da –eq $a1b1 -and $a1b1 –eq $b1c1 -and $b1c1 –eq $c1d1 -and $c1d1 –eq $d1a1 -and $d1a1 –eq $aa1 –and $aa1 –eq $bb1 -and $bb1 –eq $cc1 -and $cc1 –eq $dd1){

$p = $ab\*12

Write-Host "Perimeter = $p"

}

else { Write-Host "Invalid values"}

# 9. Використовуючи засоби PowerShell, розробити функцію для обчислення довжини відрізку у тривимірному просторі, за її допомогою обчислити периметр паралеепіпеда, заданого координатами своїх вершин

function GetLength($x1, $y1, $z1, $x2, $y2, $z2)

{

return [Math]::Sqrt([Math]::Pow(($x1-$x2),2)+[Math]::Pow(($y1-$y2),2)+[Math]::Pow(($z1-$z2),2))

}

[int]$ax = Read-Host -Prompt 'Input A(x)'

[int]$ay = Read-Host -Prompt 'Input A(y)'

[int]$az = Read-Host -Prompt 'Input A(z)'

[int]$bx = Read-Host -Prompt 'Input B(x)'

[int]$by = Read-Host -Prompt 'Input B(y)'

[int]$bz = Read-Host -Prompt 'Input B(z)'

[int]$cx = Read-Host -Prompt 'Input C(x)'

[int]$cy = Read-Host -Prompt 'Input C(y)'

[int]$cz = Read-Host -Prompt 'Input C(z)'

[int]$dx = Read-Host -Prompt 'Input D(x)'

[int]$dy = Read-Host -Prompt 'Input D(y)'

[int]$dz = Read-Host -Prompt 'Input D(z)'

[int]$ax1 = Read-Host -Prompt 'Input A1(x)'

[int]$ay1 = Read-Host -Prompt 'Input A1(y)'

[int]$az1 = Read-Host -Prompt 'Input A1(z)'

[int]$bx1 = Read-Host -Prompt 'Input B1(x)'

[int]$by1 = Read-Host -Prompt 'Input B1(y)'

[int]$bz1 = Read-Host -Prompt 'Input B1(z)'

[int]$cx1 = Read-Host -Prompt 'Input C1(x)'

[int]$cy1 = Read-Host -Prompt 'Input C1(y)'

[int]$cz1 = Read-Host -Prompt 'Input C1(z)'

[int]$dx1 = Read-Host -Prompt 'Input D1(x)'

[int]$dy1 = Read-Host -Prompt 'Input D1(y)'

[int]$dz1 = Read-Host -Prompt 'Input D1(z)'

$ab = GetLength $ax $ay $az $bx $by $dz

$bc = GetLength $bx $by $dz $cx $cy $cz

$cd = GetLength $cx $cy $cz $dx $dy $dz

$da = GetLength $dx $dy $dz $ax $ay $az

$a1b1 = GetLength $ax1 $ay1 $az1 $bx1 $by1 $dz1

$b1c1 = GetLength $bx1 $by1 $dz1 $cx1 $cy1 $cz1

$c1d1 = GetLength $cx1 $cy1 $cz1 $dx1 $dy1 $dz1

$d1a1 = GetLength $dx1 $dy1 $dz1 $ax1 $ay1 $az1

$aa1 = GetLength $ax $ay $az $ax1 $ay1 $az1

$bb1 = GetLength $bx $by $dz $b1x $by1 $bz1

$cc1 = GetLength $cx $cy $cz $cx1 $cy1 $cz1

$dd1 = GetLength $dx $dy $dz $dx1 $dy1 $dz1

$p = $ab + $bc + $cd + $da + $a1b1 + $b1c1 + $c1d1 + $d1a1 + $aa1 + $bb1 + $cc1 + $dd1

Write-Host "Perimeter = $p"

# 10.Використовуючи засоби PowerShell, розробити функцію для обчислення довжини відрізка у тривимірному просторі. За її допомогою обчислити об’єм куба, заданої координатами своїх вершин.

class Point

{

[double] $X;

[double] $Y;

[double] $Z;

}

function GetDistance([Point] $startPoint, [Point] $endPoint)

{

return [Math]::Sqrt(

[Math]::Pow(($startPoint.X - $endPoint.X), 2) +

[Math]::Pow(($startPoint.Y - $endPoint.Y), 2) +

[Math]::Pow(($startPoint.Z - $endPoint.Z), 2)

);

}

$a = New-Object Point;

$b = New-Object Point;

$c = New-Object Point;

$d = New-Object Point;

$a1 = New-Object Point;

$b1 = New-Object Point;

$c1 = New-Object Point;

$d1 = New-Object Point;

$a.X = Read-Host -Prompt 'Input A(x)';

$a.Y = Read-Host -Prompt 'Input A(y)';

$a.Z = Read-Host -Prompt 'Input A(z)';

$b.X = Read-Host -Prompt 'Input B(x)';

$b.Y = Read-Host -Prompt 'Input B(y)';

$b.Z = Read-Host -Prompt 'Input B(z)';

$c.X = Read-Host -Prompt 'Input C(x)';

$c.Y = Read-Host -Prompt 'Input C(y)';

$c.Z = Read-Host -Prompt 'Input C(z)';

$d.X = Read-Host -Prompt 'Input D(x)';

$d.Y = Read-Host -Prompt 'Input D(y)';

$d.Z = Read-Host -Prompt 'Input D(z)';

$a1.X = Read-Host -Prompt 'Input A1(x)';

$a1.Y = Read-Host -Prompt 'Input A1(y)';

$a1.Z = Read-Host -Prompt 'Input A1(z)';

$b1.X = Read-Host -Prompt 'Input B1(x)';

$b1.Y = Read-Host -Prompt 'Input B1(y)';

$b1.Z = Read-Host -Prompt 'Input B1(z)';

$c1.X = Read-Host -Prompt 'Input C1(x)';

$c1.Y = Read-Host -Prompt 'Input C1(y)';

$c1.Z = Read-Host -Prompt 'Input C1(z)';

$d1.X = Read-Host -Prompt 'Input D1(x)';

$d1.Y = Read-Host -Prompt 'Input D1(y)';

$d1.Z = Read-Host -Prompt 'Input D1(z)';

$a\_b = GetDistance $a $b;

$b\_c = GetDistance $b $c;

$c\_d = GetDistance $c $d;

$d\_a = GetDistance $d $a;

$a1\_b1 = GetDistance $a1 $b1;

$b1\_c1 = GetDistance $b1 $c1;

$c1\_d1 = GetDistance $c1 $d1;

$d1\_a1 = GetDistance $d1 $a1;

$a\_a1 = GetDistance $a $a1;

$b\_b1 = GetDistance $b $b1;

$c\_c1 = GetDistance $c $c1;

$d\_d1 = GetDistance $d $d1;

if (

$a\_b -eq $b\_c -and $a\_b -eq $c\_d -and $a\_b -eq $d\_a -and

$a\_b -eq $a1\_b1 -and $a\_b -eq $b1\_c1 -and $a\_b -eq $c1\_d1 -and $a\_b -eq $d1\_a1 -and

$a\_b -eq $a\_a1 -and $a\_b -eq $b\_b1 -and $a\_b -eq $c\_c1 -and $a\_b -eq $d\_d1

)

{ Write-Host ('Volume: ' + ($d\_a \* $a\_b \* $a\_a1)); }

else{ Write-Host 'Invalid values.'; }

pause;

# 11.Використовуючи засоби PowerShell, розробити функцію для обчислення довжини відрізка у тривимірному просторі. За її допомогою обчислити об’єм прямокутного паралелепіпеда, заданої координатами своїх вершин.

class Point

{

[double] $X;

[double] $Y;

[double] $Z;

}

function GetDistance([Point] $startPoint, [Point] $endPoint)

{

return [Math]::Sqrt(

[Math]::Pow(($startPoint.X - $endPoint.X), 2) +

[Math]::Pow(($startPoint.Y - $endPoint.Y), 2) +

[Math]::Pow(($startPoint.Z - $endPoint.Z), 2)

);

}

$a = New-Object Point;

$b = New-Object Point;

$c = New-Object Point;

$d = New-Object Point;

$a1 = New-Object Point;

$b1 = New-Object Point;

$c1 = New-Object Point;

$d1 = New-Object Point;

$a.X = Read-Host -Prompt 'Input A(x)';

$a.Y = Read-Host -Prompt 'Input A(y)';

$a.Z = Read-Host -Prompt 'Input A(z)';

$b.X = Read-Host -Prompt 'Input B(x)';

$b.Y = Read-Host -Prompt 'Input B(y)';

$b.Z = Read-Host -Prompt 'Input B(z)';

$c.X = Read-Host -Prompt 'Input C(x)';

$c.Y = Read-Host -Prompt 'Input C(y)';

$c.Z = Read-Host -Prompt 'Input C(z)';

$d.X = Read-Host -Prompt 'Input D(x)';

$d.Y = Read-Host -Prompt 'Input D(y)';

$d.Z = Read-Host -Prompt 'Input D(z)';

$a1.X = Read-Host -Prompt 'Input A1(x)';

$a1.Y = Read-Host -Prompt 'Input A1(y)';

$a1.Z = Read-Host -Prompt 'Input A1(z)';

$b1.X = Read-Host -Prompt 'Input B1(x)';

$b1.Y = Read-Host -Prompt 'Input B1(y)';

$b1.Z = Read-Host -Prompt 'Input B1(z)';

$c1.X = Read-Host -Prompt 'Input C1(x)';

$c1.Y = Read-Host -Prompt 'Input C1(y)';

$c1.Z = Read-Host -Prompt 'Input C1(z)';

$d1.X = Read-Host -Prompt 'Input D1(x)';

$d1.Y = Read-Host -Prompt 'Input D1(y)';

$d1.Z = Read-Host -Prompt 'Input D1(z)';

$a\_b = GetDistance $a $b;

$b\_c = GetDistance $b $c;

$c\_d = GetDistance $c $d;

$d\_a = GetDistance $d $a;

$a1\_b1 = GetDistance $a1 $b1;

$b1\_c1 = GetDistance $b1 $c1;

$c1\_d1 = GetDistance $c1 $d1;

$d1\_a1 = GetDistance $d1 $a1;

$a\_a1 = GetDistance $a $a1;

$b\_b1 = GetDistance $b $b1;

$c\_c1 = GetDistance $c $c1;

$d\_d1 = GetDistance $d $d1;

if (

$a\_b -eq $c\_d -and $a\_b -eq $a1\_b1 -and $a\_b -eq $c1\_d1 -and

$d\_a -eq $d1\_a1 -and $d\_a -eq $b\_c -and $d\_a -eq $b1\_c1 -and

$a\_a1 -eq $b\_b1 -and $a\_a1 -eq $c\_c1 -and $a\_a1 -eq $d\_d1

)

{Write-Host ('Volume: ' + ($d\_a \* $a\_b \* $a\_a1));}

else{Write-Host 'Invalid values.';}

pause;

# 12. Використовуючи засоби PowerShell, розробити функцію для визначення кількості цифр у файлі.

$a = $b = $c = $d = $q = $e = $f = $g = $k = $l = $m = 0

switch ([string[]][char[]](gc "Test.txt"))

{

"1" {$b++}

"2" {$c++}

"3" {$d++}

"4" {$q++}

"5" {$e++}

"6" {$f++}

"7" {$g++}

"8" {$k++}

"9" {$l++}

"0" {$m++}

} -join ""

$a = $b + $c + $d + $q + $e + $f + $g + $k + $l + $m

"Кількість цифр = "+$a

pause

# 13. Використовуючи засоби PowerShell, розробити функцію для визначення кількості англійських літер у заданому файлі

function Letters

{

$text = Get-Content F:\test.txt

$eng\_mask = @('a', 'b','c','d','e','f','g','h','j','k','l','n','o','p','q','r','s','t','u','v','w','x','y','z')

$eng\_letters = 0

foreach($letter in $text.ToCharArray()){

if($eng\_mask -match $letter){

$eng\_letters++

}

}

Write-Host $eng\_letters

}

Letters

# 14. Використовуючи засоби PowerShell, розробити функцію для визначення кількості розділових знаків у файлі.

$a = $b = $c = $d = $q = $e = 0

switch ([string[]][char[]](gc " Test.txt"))

{

" " {$b++}

"," {$c++}

"." {$d++}

"?" {$q++}

"!" {$e++}

} -join ""

" - "+$b

", - "+$c

". - "+$d

"? - "+$q

"! - "+$e

$a = $b + $c + $d + $q + $e

"Кількість розділових знаків = "+$a

pause

# 15. Використовуючи засоби PowerShell, розробити функцію для визначення кількості українських літер у заданому файлі

function Letters

{

$text = Get-Content F:\test.txt

$ukr\_mask = @('а','б','в','г','д','е','є','ж','з','и','і','ї','й','к','л','м','н','о','п','р','с','т','у','ф','х','ц','ч','ш','щ','ь','ю','я')

$ukr\_letters = 0

foreach($letter in $text.ToCharArray()){

if($ukr\_mask -match $letter){

$ukr\_letters ++

}

}

Write-Host $ukr\_letters

}

Letters