

1. ВИКОРИСТОВУЮЧИ ЗАСОБИ POWESHELL РОЗРОБИТИ РЕКУРСИВНУ ФУНКЦІЮ ДЛЯ ОБЧИСЛЕННЯ ФАКТОРІАЛА ($N!$) НАТУРАЛЬНОГО ЧИСЛА. ЗА ЇЇ ДОПОМОГОЮ ОБЧИСЛИТИ ВЕЛИЧИНУ Cnm
2. ВИКОРИСТОВУЮЧИ ЗАСОБИ POWERSHELL, РОЗРОБИТИ РЕКУРСИВНУ ФУНКЦІЮ ДЛЯ ОБЧИСЛЕННЯ ФАКТОРІАЛА ($N!$) НАТУРАЛЬНОГО ЧИСЛА. ЗА ЇЇ ДОПОМОГОЮ ОБЧИСЛИТИ ВЕЛИЧИНУ : $(24!) - (13) + (20!)$.
3. ВИКОРИСТОВУЮЧИ ЗАСОБИ POWERSHELL, РОЗРОБИТИ РЕКУРСИВНУ ФУНКЦІЮ ДЛЯ ОБЧИСЛЕННЯ ФАКТОРІАЛА ($N!$) НАТУРАЛЬНОГО ЧИСЛА. ЗА ЇЇ ДОПОМОГОЮ ОБЧИСЛИТИ ВЕЛИЧИНУ: $(N! + 34)! + (N - 3)! - (2N!)$.
4. ВИКОРИСТОВУЮЧИ ЗАСОБИ POWERSHELL, РОЗРОБИТИ ФУНКЦІЮ ДЛЯ ОБЧИСЛЕННЯ ДОВЖИНИ ВІДРІЗКУ НА ПЛОЩИНІ, ЗА ЇЇ ДОПОМОГОЮ ОБЧИСЛИТИ ПЕРИМЕТР ТРИКУТНИКА, ЗАДАНОГО КООРДИНАТАМИ СВОЇХ ВЕРШИН
5. ВИКОРИСТОВУЮЧИ ЗАСОБИ POWERSHELL, РОЗРОБИТИ ФУНКЦІЮ ДЛЯ ОБЧИСЛЕННЯ ДОВЖИНИ ВІДРІЗКА НА ПЛОЩИНІ. ЗА ЇЇ ДОПОМОГОЮ ОБЧИСЛИТИ ПЕРИМЕТР ЧОТИРИКУТНИКА, ЗАДАНОГО КООРДИНАТАМИ СВОЇХ ВЕРШИН.
6. ВИКОРИСТОВУЮЧИ ЗАСОБИ POWERSHELL, РОЗРОБИТИ ФУНКЦІЮ ДЛЯ ОБЧИСЛЕННЯ ДОВЖИНИ ВІДРІЗКА НА ПЛОЩИНІ. ЗА ЇЇ ДОПОМОГОЮ ОБЧИСЛИТИ ПЕРИМЕТР П'ЯТИКУТНИКА, ЗАДАНОГО КООРДИНАТАМИ СВОЇХ ВЕРШИН.
- 7.ВИКОРИСТОВУЮЧИ ЗАСОБИ POWERSHELL, РОЗРОБИТИ ФУНКЦІЮ ДЛЯ ОБЧИСЛЕННЯ ДОВЖИНИ ВІДРІЗКА У ТРИВИМІРНОМУ ПРОСТОРІ. ЗА ЇЇ ДОПОМОГОЮ ОБЧИСЛИТИ ПЕРИМЕТР ТРИКУТНОЇ ПІРАМІДИ, ЗАДАНОЇ КООРДИНАТАМИ СВОЇХ ВЕРШИН.
- 8.ВИКОРИСТОВУЮЧИ ЗАСОБИ POWERSHELL, РОЗРОБИТИ ФУНКЦІЮ ДЛЯ ОБЧИСЛЕННЯ ДОВЖИНИ ВІДРІЗКА У ТРИВИМІРНОМУ ПРОСТОРІ. ЗА ЇЇ ДОПОМОГОЮ ОБЧИСЛИТИ ПЕРИМЕТР КУБА, ЗАДАНОЇ КООРДИНАТАМИ СВОЇХ ВЕРШИН.
9. ВИКОРИСТОВУЮЧИ ЗАСОБИ POWERSHELL, РОЗРОБИТИ ФУНКЦІЮ ДЛЯ ОБЧИСЛЕННЯ ДОВЖИНИ ВІДРІЗКУ У ТРИВИМІРНОМУ ПРОСТОРІ, ЗА ЇЇ ДОПОМОГОЮ ОБЧИСЛИТИ ПЕРИМЕТР ПАРАЛЕЛЕПЕДА, ЗАДАНОГО КООРДИНАТАМИ СВОЇХ ВЕРШИН
- 10.ВИКОРИСТОВУЮЧИ ЗАСОБИ POWERSHELL, РОЗРОБИТИ ФУНКЦІЮ ДЛЯ ОБЧИСЛЕННЯ ДОВЖИНИ ВІДРІЗКА У ТРИВИМІРНОМУ ПРОСТОРІ. ЗА ЇЇ ДОПОМОГОЮ ОБЧИСЛИТИ ОБ'ЄМ КУБА, ЗАДАНОЇ КООРДИНАТАМИ СВОЇХ ВЕРШИН.
- 11.ВИКОРИСТОВУЮЧИ ЗАСОБИ POWERSHELL, РОЗРОБИТИ ФУНКЦІЮ ДЛЯ ОБЧИСЛЕННЯ ДОВЖИНИ ВІДРІЗКА У ТРИВИМІРНОМУ ПРОСТОРІ. ЗА ЇЇ ДОПОМОГОЮ ОБЧИСЛИТИ ОБ'ЄМ ПРЯМОКУТНОГО ПАРАЛЕЛЕПЕДА, ЗАДАНОЇ КООРДИНАТАМИ СВОЇХ ВЕРШИН.
12. ВИКОРИСТОВУЮЧИ ЗАСОБИ POWERSHELL, РОЗРОБИТИ ФУНКЦІЮ ДЛЯ ВИЗНАЧЕННЯ КІЛЬКОСТІ ЦИФР У ФАЙЛІ.
13. ВИКОРИСТОВУЮЧИ ЗАСОБИ POWERSHELL, РОЗРОБИТИ ФУНКЦІЮ ДЛЯ ВИЗНАЧЕННЯ КІЛЬКОСТІ АНГЛІЙСЬКИХ ЛІТЕР У ЗАДАНОМУ ФАЙЛІ
14. ВИКОРИСТОВУЮЧИ ЗАСОБИ POWERSHELL, РОЗРОБИТИ ФУНКЦІЮ ДЛЯ ВИЗНАЧЕННЯ КІЛЬКОСТІ РОЗДІЛОВИХ ЗНАКІВ У ФАЙЛІ.
15. ВИКОРИСТОВУЮЧИ ЗАСОБИ POWERSHELL, РОЗРОБИТИ ФУНКЦІЮ ДЛЯ ВИЗНАЧЕННЯ КІЛЬКОСТІ УКРАЇНСЬКИХ ЛІТЕР У ЗАДАНОМУ ФАЙЛІ

1. Використовуючи засоби PowerShell розробити рекурсивну функцію для обчислення факторіала (n!) натурального числа. За її допомогою обчислити величину C_n^m

```
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 відрізка 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 "Введіть xa:"
[float]$xa=read-host
Write-Host "Введіть ya:"
[float]$ya=read-host
Write-Host "Введіть xb:"
[float]$xb=read-host
Write-Host "Введіть yb:"
[float]$yb=read-host
Write-Host "Введіть xc:"
[float]$xc=read-host
Write-Host "Введіть yc:"
[float]$yc=read-host
Write-Host "Введіть xd:"
[float]$xd=read-host
Write-Host "Введіть yd:"
[float]$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 "Введіть ха:"
[float]$xa=read-host
Write-Host "Введіть уа:"
[float]$ya=read-host
Write-Host "Введіть xb:"
[float]$xb=read-host
Write-Host "Введіть yb:"
[float]$yb=read-host
Write-Host "Введіть xc:"
[float]$xc=read-host
Write-Host "Введіть yc:"
[float]$yc=read-host
Write-Host "Введіть xd:"
[float]$xd=read-host
Write-Host "Введіть yd:"
[float]$yd=read-host
Write-Host "Введіть xe:"
[float]$xe=read-host
Write-Host "Введіть ye:"
[float]$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 $bx1 $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 відрізка у тривимірному просторі, за її допомогою обчислити периметр паралелепіпеда, 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
$a1a1 = GetLength $ax $ay $az $ax1 $ay1 $az1
$b1b1 = GetLength $bx $by $dz $bx1 $by1 $dz1
$c1c1 = GetLength $cx $cy $cz $cx1 $cy1 $cz1
$d1d1 = GetLength $dx $dy $dz $dx1 $dy1 $dz1
```

```
$p = $ab + $bc + $cd + $da + $a1b1 + $b1c1 + $c1d1 + $d1a1 + $a1a1 + $b1b1 + $c1c1 + $d1d1
```

```
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