

POWERSHELL SCRIPTS

1. Write a script that will monitor the ram usage if the usage is more than 80% it should print an error message and if it is less than 80% it is normal.

```
RAM_usage.ps1 X
1  $memory = Get-WmiObject Win32_OperatingSystem
2  $totalMemory = $memory.TotalVisibleMemorySize
3  $freeMemory = $memory.freePhysicalMemory
4
5  $usedMemory = $totalMemory - $freeMemory
6  $usedMemoryPercentage = ($usedMemory / $totalMemory) * 100
7
8  $threshold = 80
9
10 if ($usedMemoryPercentage -gt $threshold){
11     Write-Host "Error: RAM usage is above $threshold %."
12 } else {
13     Write-Host "Normal: RAM usage is below $threshold %."
14 }
```

```
PS C:\Users\SGIDUTURI> C:\Users\SGIDUTURI\Desktop\RAM_usage.ps1
Normal: RAM usage is below 80%.

PS C:\Users\SGIDUTURI> C:\Users\SGIDUTURI\Desktop\RAM_usage.ps1
Normal: RAM usage is below 80%.

PS C:\Users\SGIDUTURI> C:\Users\SGIDUTURI\Desktop\RAM_usage.ps1
Normal: RAM usage is below 80.

PS C:\Users\SGIDUTURI> C:\Users\SGIDUTURI\Desktop\RAM_usage.ps1
Normal: RAM usage is below 80 %.

PS C:\Users\SGIDUTURI> |
```

RAM_usage.ps1* X

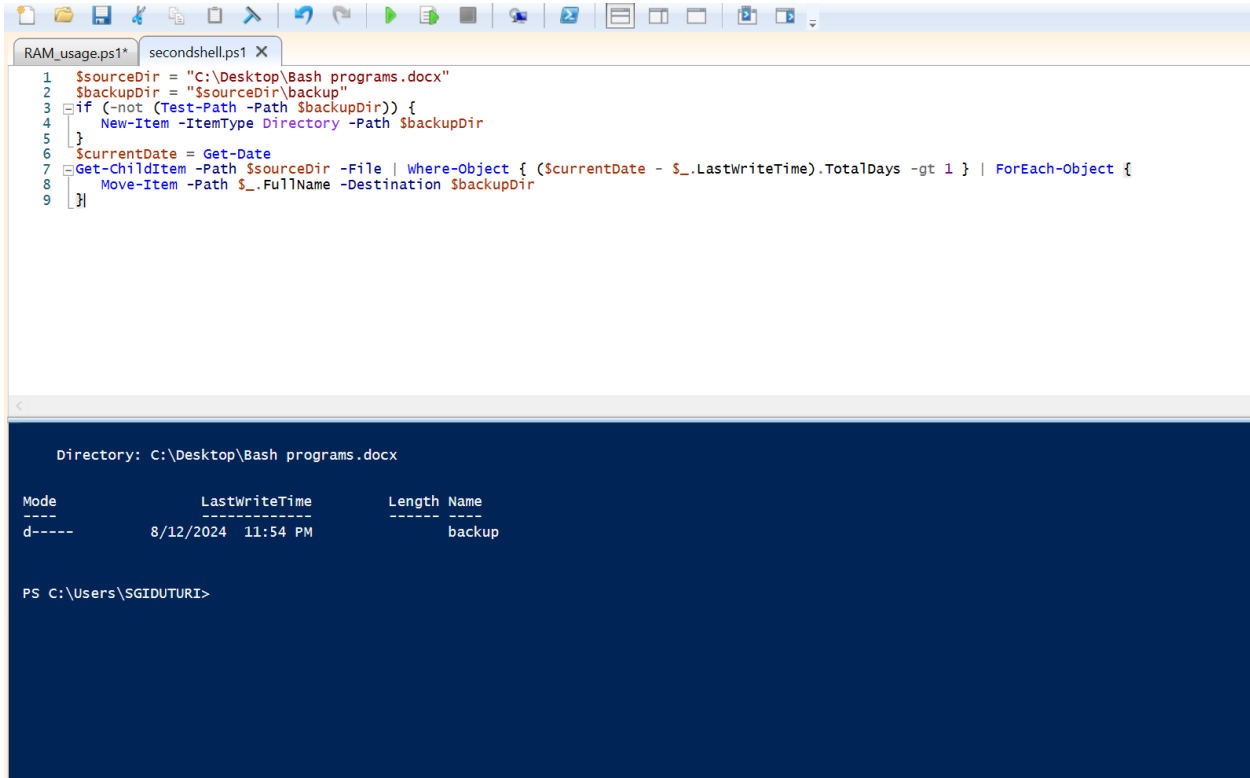
```
1 $memory = Get-WmiObject Win32_OperatingSystem
2 $totalMemory = $memory.TotalVisibleMemorySize
3 $freeMemory = $memory.FreePhysicalMemory
4
5 $usedMemory = $totalMemory - $freeMemory
6 $usedMemoryPercentage = ($usedMemory / $totalMemory) * 100
7
8 $RoundMemory = [math]::Round($usedMemoryPercentage,2)
9
10 if ($RoundMemory -gt 80){
11     Write-Host "The memory used is: $RoundMemory "
12     Write-Host "Error: RAM usage is in limit ."
13 } else {
14     Write-Host "The memory used is: $RoundMemory"
15     Write-Host "Normal: RAM usage is below limit."
16 }
```

```
PS C:\Users\SGIDUTURI> C:\Users\SGIDUTURI\Desktop\RAM_usage.ps1
The memory used is: 42.58
Normal: RAM usage is below limit.

PS C:\Users\SGIDUTURI> C:\Users\SGIDUTURI\Desktop\RAM_usage.ps1
The memory used is: 42.44
Normal: RAM usage is below limit.

PS C:\Users\SGIDUTURI>
```

1. Write a script to identify the files older than 1 day and move them into a folder named backup.



```
1 $sourceDir = "C:\Desktop\Bash programs.docx"
2 $backupDir = "$sourceDir\backup"
3 if (-not (Test-Path -Path $backupDir)) {
4     New-Item -ItemType Directory -Path $backupDir
5 }
6 $currentDate = Get-Date
7 Get-ChildItem -Path $sourceDir -File | Where-Object { ($currentDate - $_.LastWriteTime).TotalDays -gt 1 } | ForEach-Object {
8     Move-Item -Path $_.FullName -Destination $backupDir
9 }
```

Directory: C:\Desktop\Bash programs.docx

Mode	LastWriteTime	Length	Name
d-----	8/12/2024 11:54 PM		backup

PS C:\Users\SGIDUTURI>