

## GRUPA 2

**1. Wyświetl spośród wszystkich interfejsów sieciowych wartość „VlanID” tylko tych, których nazwa zawiera „Ethernet”**

```
Get-NetAdapter -name "ethernet"
```

**2. Posortuj listę procesów wg „TotalProcessorTime” (właściwość) które odwołuje się do pełnej nazwy właściwości**

```
Get-Process | sort TotalProcessorTime
```

**3. Jaką datę ma certyfikat, który najwcześniej wygaśnie (zwróć uwagę na właściwość NotAfter)**

```
Get-ChildItem Cert:\ -Recurse | sort NotAfter | Select-Object -First 1 -Property NotAfter
```

**4. Wyświetl tylko unikalne „Source” występujące wśród ostatnich 4 zdarzeń.**

```
Get-EventLog -LogName System -Newest 4 | Select-Object -Property Source -Unique
```

**5. PagedMemorySize64 - ilość pamięci wirtualnej przydzielona w pliku stronicowania pamięci wirtualnej dla skojarzonego procesu. Wartość PM jest podawana w bajtach. Przelicz wartość na MB. Wyświetl tylko 2 z najbardziej obciążających PM procesy**

```
Get-Process | sort pm | Select-Object -First 2 | ft name, @{ n="Rozmiar w KB";e={$_.pm/1MB} }
```

**6. Podaj polecenie który ma zapisać raport do pliku C:\Users\raport\_proc.txt o procesach uruchomionych w danej chwili i ilości wykorzystywanej przez nie pamięci, plik nie ma być nadpisywany a kolejne porcje informacji przy każdym uruchomieniach mają być dopisywane na koniec pliku, aby zapisy się nie wymieszały na początku skryptu przed informacjami o procesach ma znaleźć się data.**

```
Get-Date|Out-File -FilePath C:\Users\raport_proc.txt -Append ; Get-Process | ft name, vm, pm |  
Out-File -FilePath C:\Users\raport_proc.txt -Append
```

**7. Wyświetl informacje o ustawieniach dysków logicznych dla użytkowników komputera w postaci tabeli, na ekranie mają się pokazać tylko nazwę i przydziały wyłączone, przydziały niekompletne, rozmiar, nazwa woluminu**

```
Get-WmiObject -Class win32_volume | ft name, driveletter, capacity, label  Select-Object  
Name, AllocationUnitSize, DriveLetter, Capacity, Label
```

**8. Korzystając z metod WMI: Wyświetl dla każdego obiektu Win32\_NetworkAdapter tylko właściwość AdapterType oraz MACAddress.**

```
Get-WmiObject -Class Win32_NetworkAdapter | ft adaptertype, macaddress
```

**9. Korzystając z metod CIM: Zbadaj, czy komputer na jakim pracujesz jest wirtualny. W tym celu wyświetl klasę Win32\_BIOS**

```
Get-CimInstance -ClassName Win32_BIOS
```

**10. Korzystając z metod WMI i CIM: Korzystając z klasy Win32\_UserProfile wyświetl informacji o wszystkich profilach użytkowników na komputerze lokalnym**

```
Get-CimInstance Win32_UserProfile; Get-WmiObject -Class Win32_UserProfile
```

**11. Korzystając z metody ReleaseDHCPLease zwolnij adres IP uzyskany z serwera DHCP dla wybranego przez ciebie interfejsu sieciowego**

```
Get-WmiObject -Class Win32_NetworkAdapterConfiguration| Where-Object { $_.Description -  
eq "wybrany interfejs"} | foreach ReleaseDHCPLease
```

**12. Korzystając z metody RenewDHCPLease pobierz adres IP z serwera DHCP ponownie**

```
Get-WmiObject -Class Win32_NetworkAdapterConfiguration | Where-Object { $_.Description -eq "wybrany interfejs"} | foreach RenewDHCPLeas
```

13. Utwórz obiekt sesji oparty o protokół DCOM. Utwórz obiekt sesji do wybranego komputera z wykorzystaniem utworzonego obiektu sesji. Korzystając z obiektu sesji wyświetl informacje o procesorach dostępnych na zdalnym komputerze.

```
$opt=New-CimSessionOption -Protocol Dcom
```

```
$s=New-CimSession -ComputerName __nazwaPC__ -SessionOption $opt
```

```
Get-CimInstance -ClassName Win32_Processor -CimSession $s
```