

1. Zmienne Napisz skrypt wymagający podania długości i wysokości trójkąta w argumencie. Oblicza pole powierzchni trójkąta i wyświetli wynik na ekranie.

```
1 param(  
2     [Int]$dlugosc,  
3     [Int]$wysokosc)  
4     $wynik = ($dlugosc * $wysokosc) / 2  
5     $wynik
```

```
PS C:\Users\Patryk\Desktop\skrypty> .\Tab6.ps1 5 8  
20
```

2. Instrukcje warunkowe Napisz skrypt, który zapyta użytkownika o liczbę i sprawdzi, czy jest większa od 10. Jeśli tak, wyświetli komunikat "Liczba jest większa od 10", w przeciwnym razie wyświetli komunikat "Liczba jest mniejsza od 10".

```
1 [Int]$liczba = Read-Host -Prompt "Podaj liczbę: "  
2 if ($liczba -gt 10) {  
3     Write-Host "Liczba jest większa od 10"  
4 } else {  
5     Write-Host "Liczba jest mniejsza/równa 10"  
6 }
```

```
PS C:\Users\Patryk\Desktop\skrypty> C:\Users\Patryk\Desktop\skrypty>  
Podaj liczbę: : 15  
Liczba jest większa od 10  
  
PS C:\Users\Patryk\Desktop\skrypty> C:\Users\Patryk\Desktop\skrypty>  
Podaj liczbę: : 5  
Liczba jest mniejsza/równa 10
```

3. Operatory logiczne Napisz skrypt, który poprosi użytkownika o podanie nazwy użytkownika i hasła, a następnie sprawdzi, czy podane dane są poprawne (np. nazwa użytkownika to "admin" i hasło to "password"). Zastosuj maskowanie wprowadzanego hasła

```
1 $user = Read-Host -Prompt "Podaj nazwe uzytkownika: "
2 $password = Read-Host -Prompt "Podaj haslo: " -AsSecureString
3 $password = [System.Runtime.InteropServices.Marshal]::PtrToStringAuto([System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($password))
4
5 if($user -eq "admin" -and $password -eq "password") {
6     Write-Host "Poprawne dane!"
7 } else {
8     Write-Host "Niepoprawne dane!"
9 }
```

PS C:\Users\Patryk\Desktop\skrypty> C:\Users\Patryk\Desktop\skrypty\lab6.ps1  
Podaj nazwe uzytkownika: : admin  
Poprawne dane!

4. Pętle Napisz skrypt generujący 10 adresów IP za pomocą pętli, które będą różniły się czwartym oktetem rosnąco w adresie 192.168.1.X

```
1 $ip = @()
2 for($i = 1; $i -lt 11; $i++){
3     $ip += "192.168.1.$i"
4 }
5 $ip
```

PS C:\Users\Patryk\Desktop\skrypty> C:\Users\Patryk\Desktop\skrypty\lab6.ps1  
192.168.1.1  
192.168.1.2  
192.168.1.3  
192.168.1.4  
192.168.1.5  
192.168.1.6  
192.168.1.7  
192.168.1.8  
192.168.1.9  
192.168.1.10

5. Funkcje Utwórz skrypt wyświetlający informacje o systemie: Aktualna data, Wersja systemu, Użytkownik, Adres IP. Każda informacja powinna znajdować się w osobnej funkcji. Użyj jednej zmiennej z nazwą komputera w każdej funkcji. Przykład: „Aktualna data na (Nazwa komputera) to X”

```
1 function date{
2     Write-Host "Aktualna data na $env:COMPUTERNAME to:" (Get-Date)
3 }
4
5
6
7 function wersja_systemu{
8     Write-Host "Aktualna wersja systemu na $env:COMPUTERNAME to:" (Get-ComputerInfo | Select-Object -ExpandProperty OsVersion)
9 }
10
11 function uzytkownik{
12     Write-Host "Aktualny uzytkownik na $env:COMPUTERNAME to:" ($env:USERNAME)
13 }
14
15 function adres_ip {
16     Write-Host "Adres IP na $env:COMPUTERNAME to:" (Get-NetIPAddress -InterfaceIndex 2 -AddressFamily IPv4 | Select -ExpandProperty IPAddress)
17 }
18 date
19 wersja_systemu
20 uzytkownik
21 adres_ip
```

PS C:\Users\Patryk\Desktop\skrypty> C:\Users\Patryk\Desktop\skrypty\lab6.ps1  
Aktualna data na DESKTOP-QM1R6ER to: 10.04.2025 16:50:26  
Aktualna wersja systemu na DESKTOP-QM1R6ER to: 10.0.19042  
Aktualny uzytkownik na DESKTOP-QM1R6ER to: Patryk  
Adres IP na DESKTOP-QM1R6ER to: 192.168.131.129

6.

```
1 $zawartosc = gc .\input_file.csv
2 for($i = 1;$i -lt $zawartosc.Length; $i++){
3
4     $nazwa, $status = $zawartosc[$i].Split(";")
5     $info = ""
6     if ($status -eq "True"){
7
8         if ($nazwa -eq "Computername") { $info = "CsName" }
9         elseif($nazwa -eq "Manufacturer") { $info = "CsManufacturer" }
10        elseif($nazwa -eq "Model") { $info = "CsModel" }
11        elseif($nazwa -eq "SerialNumber") {
12            Write-Host "SerialNumber:" (Get-WmiObject -Class Win32_BIOS | Select-Object -ExpandProperty SerialNumber)
13            continue
14        }
15        elseif($nazwa -eq "CpuName") { $info = "CsProcessors" }
16        elseif($nazwa -eq "RAM") {
17            $ram = Get-ComputerInfo | Select-Object -ExpandProperty OsTotalVisibleMemorySize
18            $ram = [math]::Ceiling($ram / 1048576)
19            Write-Host "RAM: ${ram}GB"
20            continue
21        }
22
23        Write-Host "${nazwa}:" (Get-ComputerInfo | Select-Object -ExpandProperty $info )
24    }
25 }
```

PS C:\Users\Patryk\Desktop\skrypty> C:\Users\Patryk\Desktop\skrypty\lab6.ps1  
Computername: DESKTOP-QM1R6ER  
Manufacturer: VMware, Inc.  
Model: VMware20,1  
SerialNumber: VMware-56 4d 8a f9 5e 0e 8d 52-7a c7 80 6f fb 9e a7 ca  
RAM: 5GB  
PS C:\Users\Patryk\Desktop\skrypty>

# Dodatkowe

```
1 $zawartosc = gc .\input_file.csv
2 $plik_wyjsciowy = "ComputerReport_$(Get-Date -Format yyyyMMdd)-$(Get-Date -Format HHmmss).txt"
3 for($i = 1;$i -lt $zawartosc.Length; $i++){
4
5     $nazwa, $status = $zawartosc[$i].Split(";")
6     $info = ""
7     if ($status -eq "True"){
8
9         if ($nazwa -eq "Computername") { $info = "CsName" }
10        elseif($nazwa -eq "Manufacturer") { $info = "CsManufacturer" }
11        elseif($nazwa -eq "Model") { $info = "CsModel" }
12        elseif($nazwa -eq "SerialNumber") {
13            Write-Output "SerialNumber: $(Get-WmiObject -Class Win32_BIOS | Select-Object -ExpandProperty SerialNumber)" >> $plik_wyjsciowy
14            continue
15        }
16        elseif($nazwa -eq "CpuName") { $info = "CsProcessors" }
17        elseif($nazwa -eq "RAM") {
18            $ram = Get-ComputerInfo | Select-Object -ExpandProperty OsTotalVisibleMemorySize
19            $ram = [math]::Ceiling($ram / 1048576)
20            Write-Output "RAM: ${ram}GB" >> $plik_wyjsciowy
21            continue
22        }
23        Write-Output "${nazwa}: $(Get-ComputerInfo | Select-Object -ExpandProperty $info )" >> $plik_wyjsciowy
24    }
25 }
26 }
27 }
28 }
29 Write-Host "Raport został zapisany w pliku: $(Get-Location | Select-Object -ExpandProperty Path)\$plik_wyjsciowy"
```

```
PS C:\Users\Patryk\Desktop\skrypty> C:\Users\Patryk\Desktop\skrypty\lab6.ps1
Raport został zapisany w pliku: C:\Users\Patryk\Desktop\skrypty\ComputerReport_20250412-223437.txt
PS C:\Users\Patryk\Desktop\skrypty>
```

ComputerReport\_20250412-223437.txt — Notatnik

Plik Edycja Format Widok Pomoc

Computername: DESKTOP-GM1R6ER  
Manufacturer: VMware, Inc.  
Model: VMware20,1  
SerialNumber: VMware-56 4d 8a f9 5e 0e 8d 52-7a c7 80 6f fb 9e a7 ca  
RAM: 5GB

Lin 1, kol 1      100%      Windows (CRLF)      UTF-16 LE