Boletin 1 Powershell

1.- Realizar un programa en Powershell que pida un número e devolva si
é primo ou non . Utiliza funcións na medida do posible. [Opcional] O programa tenos que permitir volver a facer ese cálculo unha e outra vez si así o desexamos.

```powershell
function Es-Primo {
    param (
        [int]$Numero
    )

    if ($Numero -lt 2) {
        return $false
    }

    for ($i = 2; $i -le [math]::Sqrt($Numero); $i++) {
        if ($Numero % $i -eq 0) {
            return $false
        }
    }

    return $true
}

function Principal {
    do {
        Write-Host "Introduce un número para verificar si es primo:"
        $numeroIngresado = Read-Host "Número"

        if (-not [int]::TryParse($numeroIngresado, [ref]$null)) {
            Write-Host "Por favor, introduce un número válido."
            continue
        }

        if (Es-Primo -Numero $numeroIngresado) {
            Write-Host "$numeroIngresado es un número primo."
        } else {
            Write-Host "$numeroIngresado no es un número primo."
        }

        Write-Host "¿Quieres verificar otro número? (s/n):"
        $respuesta = Read-Host
    } while ($respuesta -match '^(s|S)$')
}

```

2.- Escribir un xogo de adiviñanza. O programa pedirá ao usuario dous números (o número inferior e o número superior). O programa obterá, a continuación, un número aleatorio entre eses dous números, e o usuario deberá adiviñalo. Cada vez que o usuario introduce un número, o programa lle dice si é maior ou menor. Ao final, o programa indica o número de intentos utilizado.

```powershell
Write-Host "Benvido ao xogo da adiviñanza máxica!"
Write-Host "Primeiro, necesitamos establecer os límites do xogo."


$límiteInferior = Read-Host "Introduce o límite inferior"
$límiteSuperior = Read-Host "Introduce o límite superior"


if (-not ([int]::TryParse($límiteInferior, [ref]$null) -and [int]::TryParse($límiteSuperior, [ref]$null))) {
    Write-Host " Erro: Ambos os límites deben ser números válidos."
    exit
}

$límiteInferior = [int]$límiteInferior
$límiteSuperior = [int]$límiteSuperior


if ($límiteInferir -ge $límiteSuperior) {
    Write-Host " Erro: O límite inferior debe ser menor que o límite superior."
    exit
}


$numeroAleatorio = Get-Random -Minimum $límiteInferior -Maximum ($límiteSuperior + 1)
$intentos = 0

Write-Host "Pensei nun número entre $límiteInferior e $límiteSuperior."
Write-Host "¿Poderás adiviñalo? ¡Imos aló!"

while ($true) {
    $intentoUsuario = Read-Host "Introduce a túa adiviñanza"
    $intentos++

    if (-not [int]::TryParse($intentoUsuario, [ref]$null)) {
        Write-Host " Por favor, introduce un número válido."
        continue
    }

    $intentoUsuario = [int]$intentoUsuario


    if ($intentoUsuario -eq $numeroAleatorio) {
        Write-Host " Parabéns! Adiviñaches o número $numeroAleatorio en $intentos intento(s)."
        break
    } elseif ($intentoUsuario -lt $numeroAleatorio) {
        Write-Host " O número que pensaches é máis pequeno. Inténtao de novo."
    } else {
        Write-Host " O número que pensaches é máis grande. Inténtao de novo."
    }
}

Write-Host "Grazas por xogar ao xogo da adiviñanza máxica. ¡Volve pronto!"

```

3.- Programa que recolla unha lista de números (ata que o usuario escriba un 0) e logo devolva esa lista ordenada.

[Opcional]

```powershell
Write-Host "Benvido! Vamos a crear unha lista de números."
Write-Host "Introduce números un por un. Escribe 0 para finalizar."


$listaNumeros = @()


while ($true) {
    $numero = Read-Host "Introduce un número"


    if (-not [int]::TryParse($numero, [ref]$null)) {
        Write-Host " Por favor, introduce un número válido."
        continue
    }

    $numero = [int]$numero

    if ($numero -eq 0) {
        break
    }

    $listaNumeros += $numero
}


$listaOrdenada = $listaNumeros | Sort-Object


Write-Host "Aquí tes a lista ordenada:" |
$listaOrdenada | ForEach-Object { Write-Host $_ }

```

4.- Realizar un programa en Powershell que realice a factorización dun número enteiro.
(Por exemplo: 120 = 2 * 2 * 2 * 3 * 5)
Despois debe amosalo do seguinte xeito: $120 = 2^3 * 3^1 * 5^1$

```powershell
Write-Host "Benvido ao programa de factorización!"
Write-Host "Introduce un número enteiro para factorizar:"


$numero = Read-Host "Número"


if (-not [int]::TryParse($numero, [ref]$null)) {
    Write-Host " Por favor, introduce un número válido."
    exit
}

$numero = [int]$numero


if ($numero -le 0) {
    Write-Host " Por favor, introduce un número positivo maior que 0."
    exit
}


$factoresPrimos = @()
$divisor = 2


while ($numero -gt 1) {
    if ($numero % $divisor -eq 0) {
        $factoresPrimos += $divisor
        $numero = [math]::Floor($numero / $divisor)
    } else {
        $divisor++
    }
}


Write-Host "A factorización é:"
Write-Host ($factoresPrimos -join " * ")


```