scripting en POWERSHELL – ejercicios

Para hacer estos ejercicios necesitamos una máquina virtual con Ubuntu 20

1. Crea un script que pida por teclado el nombre, después los dos apellido, permanezca esperando durante 5 segundos y después muestre por pantalla el mensaje “Tu nombre completo es ….”

|  |
| --- |
| $nombre=Read-Host "¿Como te llamas?"  $apellidos=Read-Host "¿Cuáles son tus apellidos?"  Write-Host "Tu nombre completo es $nombre $apellidos" |

1. Crea un script que pida un número y muestre su tabla de multiplicar.

|  |
| --- |
|  |

1. Crea un script que muestre un menú con las opciones:
   * Suma
   * Resta
   * Multiplicación
   * División
   * Salir

Las 4 primeras opciones pedirán dos números y realizarán con esos números la operación correspondiente, mostrarán un mensaje con la solución (numero1 operación numero2 = resultado) y volverán a pedir que se elija una opción del menú (no hace falta volver a escribir el menú). La última opción terminará el script.

Utiliza funciones

1. Crea un script para que haga las siguientes operaciones:

* Pedirá un nombre para los archivos que quieres crear
* Pedirá la cantidad de archivos que quieres crear
* Los archivos creados tendrán el siguiente formato (se crearán en el directorio personal del usuario:

*número\_de\_archivo\_nombre\_de\_archivo.txt*

* Dentro del archivo creado tendremos la siguiente frase:

Este es el archivo número (número de archivo)

1. Un administrador debe revisar día a día si hay conexión hacia todas las máquinas contenidas en un archivo de la forma:

maquina1;IP

maquina2;IP

…

Para hacerlo puedes usar Test-Connection (ping en PowerShell)

El script cada vez que se ejecute añadirá a un archivo log (conexiones.log) con líneas similares a:

Conexiones – fecha\_del\_dia:

maquina1 - hay conexión

maquina2 : no hay conexión

…

1. Crea un script que pide el nombre de un fichero y luego visualiza un menú con tres opciones: borrar fichero o visualizar contenido (tras realizar una de las acciones se muestra el resultado y se sale del script).
2. Crea un script que a partir de una lista de nombres de usuario contenida en el archivo alumnos.txt, uno por línea, determine si el usuario existe o no. El script mostrará un listado con los usuarios que figuran en el archivo alumnos.txt pero que no existan en el sistema.
3. Crea un script que recoja como parámetro un nombre de directorio, verifica que el directorio existe y en caso de que exista muestra el contenido del directorio ordenado por tamaño, y de cada elemento debe mostrar el tamaño y el nombre.
4. Crea un script que pida un número entre 5 y 9 y en función de dicho número dibuje por pantalla un diamante como el de la siguiente figura (con tantas filas (para cada parte del diamante) como indica dicho número.

Por ejemplo, para el número 6 tendríamos la siguiente salida

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

1. Crea un script que defina un array con 10 nombres y con un bucle recorra el array para mostrar uno a uno sus elementos (hazlo con foreach, for y while)
2. Crea un script que pida un número entre 3 y 9 y en función de dicho número dibuje por pantalla un rombo como el de la siguiente figura

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dimensión: **3** | | | | |  | Dimensión: **6** | | | | | | | | | | |
|  |  | X |  |  |  |  |  |  |  |  | X |  |  |  |  |  |
|  | X |  | X |  |  |  |  |  |  | X |  | X |  |  |  |  |
| X |  |  |  | X |  |  |  |  | X |  |  |  | X |  |  |  |
|  | X |  | X |  |  |  |  | X |  |  |  |  |  | X |  |  |
|  |  | X |  |  |  |  | X |  |  |  |  |  |  |  | X |  |
|  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  | X |
|  |  |  |  |  |  |  | X |  |  |  |  |  |  |  | X |  |
|  |  |  |  |  |  |  |  | x |  |  |  |  |  | X |  |  |
|  |  |  |  |  |  |  |  |  | X |  |  |  | X |  |  |  |
|  |  |  |  |  |  |  |  |  |  | x |  | X |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |