**Ejercicios 02**

6º Se proponen varias adivinanzas que el usuario tiene que adivinar

Adivinanza 1: "Oro parece plata no es, el que no lo adivine bien tonto que es."

Respuestas correctas serán platano o plátano, escritas tanto en mayúsculas como en minúsculas (usar ToLower)

Adivinanza 2: "Soy una señora muy señoreada siempre voy en coche y siempre voy mojada"

"Respuestas correctas la lengua, lengua tanto en mayúsculas como en minúsculas

No olvidar los comentarios en el programa. Puedes poner los mensajes al usuario que consideres oportunos para hacer el programa lo más amigable posible

|  |
| --- |
| #Sexto ejercicio  #Autor: Emmanuel Ramos Biado  #Fecha de creación: 10/03/2022  <#  Se proponen varias adivinanzas que el usuario tiene que adivinar  Adivinanza 1: "Oro parece plata no es, el que no lo adivine bien tonto que es."  Respuestas correctas serán platano o plátano, escritas tanto en mayúsculas como en minúsculas (usar ToLower)  Adivinanza 2: "Soy una señora muy señoreada siempre voy en coche y siempre voy mojada"  "Respuestas correctas la lengua, lengua tanto en mayúsculas como en minúsculas  #>  Clear-Host  Write-Host "Te propongo dos adivinanzas"  Read-Host "Pulsa cualquier tecla para continuar"  Clear-Host  #Adivinanza 1  Write-Host "Adivinanza Nº 1:"  Write-Host "Oro parece plata no es, el que no lo adivine bien tonto que es."  $respuesta1 = Read-Host "Introduce tu respuesta"  if (($respuesta1 -ieq "platano") -or ($respuesta1 -ieq "plátano"))  {  Write-Host "Lo has adivinado."  }  else  {  Write-Host "No lo has adivinado."  }  Read-Host "Pulsa una tecla para continuar con la siguiente adivinanza"  Clear-Host  #Adivinanza 2  Write-Host "Adivinanza Nº 2:"  Write-Host "Soy una señora muy señoreada siempre voy en coche y siempre voy mojada."  $respuesta2 = Read-Host "Introduce tu respuesta"  if ($respuesta2 -ieq "lengua")  {  Write-Host "Lo has adivinado."  }  else  {  Write-Host "No lo has adivinado."  } |

7º Crear un programa que visualice el contenido de la variable de entorno PATH. A continuación le pregunta al usuario si desea ampliar el contenido de dicha variable. Si es que no acaba el programa y en caso contrario le pedirá una ruta. Comprobar con Test-Path si dicha ruta existe, en caso contrario sacar los mensajes apropiados. Si existe modificar el path para que añada estar ruta al final. ¡Ten cuidado de no perder el contenido actual de $PATH! Al final vuelve a mostrar el contenido de la variable de entorno $Path para ver que ha añadido la nueva ruta al final.

|  |
| --- |
| #Séptimo ejercicio  #Autor: Emmanuel Ramos Biado  #Fecha de creación: 10/03/2022  <#  Crear un programa que visualice el contenido de la variable de entorno PATH.  A continuación le pregunta al usuario si desea ampliar el contenido de dicha variable.  Si es que no acaba el programa y en caso contrario le pedirá una ruta.  Comprobar con Test-Path si dicha ruta existe, en caso contrario sacar los mensajes apropiados.  Si existe modificar el path para que añada estar ruta al final.  ¡Ten cuidado de no perder el contenido actual de $PATH! Al final vuelve a mostrar el contenido de la variable de entorno $Path para ver que ha añadido la nueva ruta al final.  #>  Clear-Host  #Muestra el contenido de la variable de entorno PATH  Get-Content Env:Path  Write-Host "" #Salto de línea  $valor = Read-Host "¿Desea ampliar el contenido de dicha variable? (S/N)"  #El usuario introduce "S"  if ($valor -ieq "s")  {  #Introducir una ruta  $ruta = Read-Host "Inserta una ruta"    #La ruta existe  if (Test-Path $ruta)  {  $env:Path = $ruta  Write-Host "La nueva ruta es: " + (Get-Content Env:Path)  }  #La ruta no existe  else  {  Write-Host "Esa ruta no existe"  }  }  #El usuario introduce "N" o cualquier otra tecla  else  {  exit  } |

8º Crea un programa que muestre un menú con los módulos de este curso. El usuario introduce uno de los módulos del menú y le dirá que profesor lo imparte. Se mostrará un menú con números y los módulos del ciclo. Si ha habido más de un profesor, se le dirá al usuario y se le preguntará si quiere saber el nombre del otro profesor.

Se quedará en el menú mientras no se pulse la opción de salir. Al dar el nombre se parará la ejecución durante 3 segundos para dar tiempo al usuario a leer el profesor. Luego volverá a salir el menú. Al pulsar la opción salir saldrá un mensaje de despedida, en el que se le deseará suerte con el curso y acabará el programa.

|  |
| --- |
| #Octavo ejercicio  #Autor: Emmanuel Ramos Biado  #Fecha de creación: 10/03/2022  <#  Crea un programa que muestre un menú con los módulos de este curso.  El usuario introduce uno de los módulos del menú y le dirá que profesor lo imparte.  Se mostrará un menú con números y los módulos del ciclo. Si ha habido más de un profesor, se le dirá al usuario y se le preguntará si quiere saber el nombre del otro profesor.  Se quedará en el menú mientras no se pulse la opción de salir.  Al dar el nombre se parará la ejecución durante 3 segundos para dar tiempo al usuario a leer el profesor.  Luego volverá a salir el menú.  Al pulsar la opción salir saldrá un mensaje de despedida, en el que se le deseará suerte con el curso y acabará el programa.  #>  Clear-Host  $numero = [int] 0  while ($numero -ne 7)  {  #Menú de selección  Write-Host "Módulos de este curso:"  Write-Host "1 - Lenguajes de marcas"  Write-Host "2 - Entornos de desarrollo"  Write-Host "3 - Bases de datos"  Write-Host "4 - Formación y orientación laboral"  Write-Host "5 - Sistemas informáticos"  Write-Host "6 - Programación"  Write-Host "7 - Salir"  $numero = Read-Host "Selecciona un número del menú"  switch ($numero)  {  1 {  Write-Host "`n`Ha seleccionado Lenguajes de marcas"  Write-Host "La profesora es Esperanza Bonillo"  Start-Sleep -s 3  Clear-Host  }  2 {  Write-Host "`n`Ha seleccionado Entornos de desarrollo"  Write-Host "El profesor es Jose Emilio Vera"  Start-Sleep -s 3  Clear-Host  }  3 {  Write-Host "`n`Ha seleccionado Base de datos"  Write-Host "El profesor se llama Alfonso Navajas"  Start-Sleep -s 3  Clear-Host  }  4 {  Write-Host "`n`Ha seleccionado Formación y orientación laboral"  Write-Host "La profesora es Ana Rosa Guerra"  Start-Sleep -s 3  Clear-Host  }  5 {  Write-Host "`n`Ha seleccionado Sistemas informáticos"  Write-Host "La profesora es Carmen Armijo"  Start-Sleep -s 3  Clear-Host  }  6 {  Write-Host "`n`Ha seleccionado Programación"  Write-Host "El profesor es Alberto López"  Start-Sleep -s 3  Write-Host "`n`En este módulo hay otra profesora. ¿Quiere saber su nombre?"  $opcion = Read-Host "Elige (s/n)"  if ($opcion -ieq "s")  {  Write-Host "`n`La otra profesora se llama Eva Concepción López"  Start-Sleep -s 3  }  Clear-Host  }  7 {  Write-Host "`n`Que tenga un buen día."  }    }  } |

9º Crea un programa que muestre cien números en orden descendente y por cada uno indica El número x es par (o impar) según proceda. Luego pregunta al usuario si desea los números pares o impares y según lo que responda mostrar solo esos números esta vez en orden ascendente. Pon los mensajes de usuario que considere pertinentes. Comenta el programa.

|  |
| --- |
| #Noveno ejercicio  #Autor: Emmanuel Ramos Biado  #Fecha de creación: 10/03/2022  <#  Crea un programa que muestre cien números en orden descendente y por cada uno indica El número x es par (o impar) según proceda.  Luego pregunta al usuario si desea los números pares o impares y según lo que responda mostrar solo esos números esta vez en orden ascendente.  Pon los mensajes de usuario que considere pertinentes. Comenta el programa.  #>  Clear-Host  #Mostrar los 100 números en orden descendiente  for ($i = 100; $i -gt 0; $i--)  {  if ($i%2 -eq 0)  {  Write-Host $i "Par"  }  if ($i%2 -ne 0)  {  Write-Host $i "Impar"  }  }  #Elegir Par o Impar  Write-Host "¿Desea los números PARES o IMPARES?"  $valor = Read-Host "Introduce par o impar"  #Pares en orden ascendente  if ($valor -ieq "par")  {  for ($i = 1; $i -lt 101; $i++)  {  if ($i%2 -eq 0)  {  Write-Host $i  }  }  }  #Impares en orden ascendente  if ($valor -ieq "impar")  {  for ($i = 1; $i -lt 101; $i++)  {  if ($i%2 -ne 0)  {  Write-Host $i  }  }  } |

10º Escribe un programa que te indique para la fecha actual si estamos o no en horario de verano. . Introduce en una variable lo siguiente:

Get-Date -Format o | ForEach-Object { $\_ -replace ":", "." }

y con el resultado de esa variable crea un directorio. Pregunta al usuario si quiere crear un fichero, si es así que te diga el nombre y lo creas vacío. Lo preguntará varias veces hasta que diga que no quiere crear más ficheros.

|  |
| --- |
| #Décimo ejercicio  #Autor: Emmanuel Ramos Biado  #Fecha de creación: 11/03/2022  <#  Escribe un programa que te indique para la fecha actual si estamos o no en horario de verano.  Introduce en una variable lo siguiente:  Get-Date -Format o | ForEach-Object { $\_ -replace ":", "." }  y con el resultado de esa variable crea un directorio.  Pregunta al usuario si quiere crear un fichero, si es así que te diga el nombre y lo creas vacío.  Lo preguntará varias veces hasta que diga que no quiere crear más ficheros.  #>  Clear-Host  #Horario de verano  $fecha = (Get-Date -DisplayHint Date)  Write-Host "Estamos a $fecha"  $horario = $fecha.IsDaylightSavingTime()  if($horario){  Write-Host "Estamos en el horario de verano"  } else{  Write-Host "No estamos en el horario de verano"  }  #Variable  $variable = Get-Date -Format o | ForEach-Object { $\_ -replace ":", "." }  New-Item -Path .\Desktop\%formato -ItemType Directory  #Crear los ficheros  do{  $valor = Read-Host "¿Quieres crear un fichero?(S/N)"  if($valor.ToLower() -eq 's'){  $nombre = Read-Host "Introduce el nombre del fichero"  New-Item .\Desktop\$nombre.txt -ItemType File  } else {  Write-Host "Hasta pronto"  }  }while ($valor.ToLower() -eq 's') |