1.- Diseña un script que cree una tabla asociativa que contenga los datos de los **usuarios normales** de un sistema Linux Ubuntu con el siguiente formato:

**login:otra\_información**

**login es el login del usuario y otra\_información es el nombre del directorio de conexión y la shell utilizada.**

Por ejemplo

***antonio: /home/antonio-/bin/bash***

***juan:/home/juan-/bin/dash***

Después, visualiza en pantalla el contenido de la tabla obtenida**.**

**#!/bin/bash**

**# Declaramos un array asociativo donde se iran guardando los usuarios normales del sistema (UID=>1000)**

**declare -A USUARIOS\_SISTEMA**

**# Guardamos en una variable el fichero que vamos a tratar, en este caso el fichero de cuentas linux /etc/passwd**

**FICHERO="/etc/passwd"**

**# Con un bucle while leemos el fichero y cada linea que va leyendo cada campo lo guarda en las variables**

**while IFS=: read -r LOGIN CLAVE ID\_USUARIO ID\_GRUPO INFO\_USUARIO HOME SHELL; do**

**if (( ID\_USUARIO >= 1000 )); then**

**USUARIOS\_SISTEMA["$LOGIN"]="$HOME $SHELL"**

**fi**

**done < "$FICHERO"**

**# Mostramos con un bucle for cada linea de login con el array asociativo. Mostramos todo el array generado con el bucle while**

**for LOGIN in "${!USUARIOS\_SISTEMA[@]}"; do**

**echo "$LOGIN: ${USUARIOS\_SISTEMA[$LOGIN]}"**

**done**

2.- Diseña un script que cree un **array** con los nombres de los servicios que estén “enabled”. Después, muestra en pantalla el nombre de estos, cada uno de ellos en una línea diferente.

¿Cuántos tenemos? ¿Tienen los arrays algún atributo o función que permita saber el tamaño del mismo?

**#!/bin/bash**

**# Declaro un array vacío donde se iran guardando los servicios que estén activos**

**SERVICIOS\_ACTIVOS=()**

**# Obtenemos una lista de servicios habilitados y filtramos solo el nombre del servicio**

**while IFS=' ' read -r SERVICIO ESTADO; do**

**if [ "$ESTADO" == "enabled" ]; then**

**SERVICIOS\_ACTIVOS+=("$SERVICIO")**

**fi**

**done < <(systemctl list-unit-files | tail -n +2 | cut -d ' ' -f1,2)**

**# Imprimimos el array que contiene los servicios habilitados**

**echo "Los servicios activos son: ${SERVICIOS\_ACTIVOS[\*]}"**

3.- Diseña un script bash que genere un fichero de 100 filas. El contenido de cada fila serán 5 valores separados por el carácter “,” .

Los valores posibles son MB, B, R, M, MM (MB es Muy Bueno, R es Regular, MM es Muy Malo). Estos valores se generarán de forma aleatoria.

**#!/bin/bash**

**FICHERO="fichero\_ramdom.txt"**

**VALORES\_POSIBLES=("MB" "B" "R" "M" "MM")**

**TAMANIO\_ARRAY=${#VALORES\_POSIBLES[@]}**

**touch "$FICHERO"**

**for VALOR in {1..100} ; do**

**VALOR\_ALEATORIO=$(($RANDOM % $TAMANIO\_ARRAY))**

**echo ${VALORES\_POSIBLES[$VALOR\_ALEATORIO]} "," ${VALORES\_POSIBLES[$VALOR\_ALEATORIO]} >> "$FICHERO"**

**done**

4.- En nuestra empresa hay mucha gente que le gusta hacer deporte y quieren empezar el año superando uno de tantos retos que circulan por las redes sociales. En concreto hay uno que consiste en hacer un ejercicio un número determinado de veces cada día: el 1 de enero, 1 repetición, el 2 de enero, 2 repeticiones, …, el 7 de marzo, 67 repeticiones y así hasta el último día del año. Los ejercicios son, de momento, **burpees**, **flexiones,** **dominadas** y **abdominales** (actualmente son 4 pero sabemos que serán más).

Se pide que realices un **script** en bash que calcule el total de repeticiones que se deben hacer en un año de ese ejercicio.

El programa solicita el año que quiere hacerlo, así como el ejercicio que quiere realizar (se debe mostrar en pantalla los que hay para ayudar al usuario). Si no introduce un ejercicio correcto, se le muestra un mensaje y se termina el programa.

Hay que tener en cuenta si el año es o no bisiesto.

***Un año bisiesto es aquel que es divisible por 4, excepto si es divisible por 100. Sin embargo, si un año es divisible por 100 y también es divisible por 400, entonces sigue siendo considerado bisiesto.***

5.- Implementa para el IES Gregorio Prieto un chat conversacional que intente solventar alguna duda a quién lo utilice.

Cuando se ejecute, aparece el mensaje:

***¡Bienvenido al chat conversacional del IES Gregorio Prieto!***

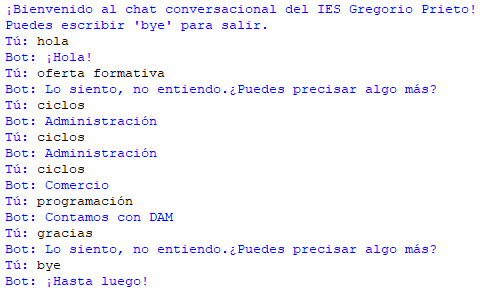
***Puedes escribir 'bye' para salir*.**

Después, aparece el prompt **Tú:**

Tecleamos lo que consideremos y el chat nos mostrará en su prompt, **Bot:** una respuesta de las que tengamos asociadas a lo que ha escrito el usuario o el mensaje **Lo siento, no entiendo. ¿Puedes precisar algo más?**

Así hasta que teclee bye.

Ejemplo de ejecución.



**#!/bin/bash**

**# Definir preguntas y respuestas**

**declare -A responses**

**responses=(**

**["hola"]="¡Hola!"**

**["oferta formativa"]="Lo siento, no entiendo. ¿Puedes precisar algo más?"**

**["ciclos"]="Tenemos ciclos formativos de grado medio y superior"**

**["programacion"]="Contamos con DAM"**

**["redes"]="Contamos con ASIR"**

**)**

**# Función para obtener respuesta**

**get\_response() {**

**local question=$1**

**echo "${responses[$question]}"**

**}**

**# Función principal del bot**

**chatbot() {**

**echo "¡Bienvenido al chat conversacional del IES Gregorio Prieto!"**

**echo \n**

**echo "Puedes escribir 'bye' para salir"**

**while true; do**

**# Leer la entrada del usuario**

**read -p "Tú: " user\_input**

**# Salir del chatbot si el usuario escribe 'salir'**

**if [[ "$user\_input" == "bye" ]]; then**

**echo "Bot: ¡Hasta luego!"**

**break**

**fi**

**# Procesar la entrada y obtener respuesta**

**response=$(get\_response "$user\_input")**

**# Responder al usuario o dar una respuesta genérica**

**if [[ -n "$response" ]]; then**

**echo "Bot: $response"**

**else**

**echo "Bot: Lo siento, no entiendo. ¿Puedes precisar algo más?."**

**fi**

**done**

**}**

**# Ejecutar el chatbot**

**chatbot**

6.- Diseña un script bash que muestre los recursos compartidos Samba en forma de página HTML

**#!/bin/bash**

**# Variable con el nombre del fichero que voy a generar**

**NOMBRE\_FICHERO="samba.html"**

**read -p "¿Dónde desea guardar el fichero $NOMBRE\_FICHERO?: " DIRECTORIO\_DESTINO**

**if [ ! -d "$DIRECTORIO\_DESTINO" ]; then**

**echo "El directorio $DIRECTORIO\_DESTINO no existe"**

**exit 1**

**fi**

**# Extraemos las líneas de carpetas compartidas**

**RECURSOS\_COMPARTIDOS=$(grep '^\[.\*\]$' /etc/samba/smb.conf)**

**# Creamos el fichero HTML**

**echo "<html>**

**<head>**

**<title>Recursos compartidos con Samba</title>**

**</head>**

**<body>**

**<h1>Carpetas compartidas con Samba en este equipo</h1>**

**<table border='1'>**

**<tr>**

**<th>Carpeta Compartida</th>**

**</tr>" > "$DIRECTORIO\_DESTINO/$NOMBRE\_FICHERO"**

**# Procesamos el contenido de los recursos compartidos línea a línea**

**while IFS= read -r LINEA; do**

**if [[ -n "$LINEA" ]]; then**

**# Escribimos en la tabla HTML**

**echo "<tr>" >> "$DIRECTORIO\_DESTINO/$NOMBRE\_FICHERO"**

**echo "<td>$LINEA</td>" >> "$DIRECTORIO\_DESTINO/$NOMBRE\_FICHERO"**

**echo "</tr>" >> "$DIRECTORIO\_DESTINO/$NOMBRE\_FICHERO"**

**fi**

**done <<< "$RECURSOS\_COMPARTIDOS"**

**# Cerramos el fichero HTML**

**echo "</table>**

**</body>**

**</html>" >> "$DIRECTORIO\_DESTINO/$NOMBRE\_FICHERO"**

**echo "Se ha creado correctamente el fichero $DIRECTORIO\_DESTINO/$NOMBRE\_FICHERO"**