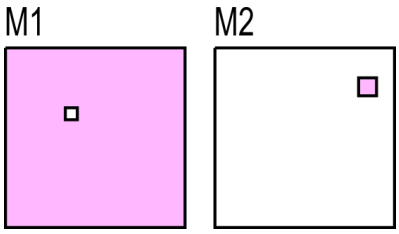
Examen de shell script. Diciembre 2022



(**I**) Escribir un guión shell denominado *matrixGen* que genere en salida estándar una matriz cuadrada de enteros aleatorios entre 0 y 9 de tamaño NxN donde N es el parámetro del guión. Los componentes de cada fila estarán separados por un espacio en blanco. Tenga en cuenta que la expresión `$(($RANDOM % 10))` genera un número aleatorio entre 0 y 9. Se invocaría, por ejemplo, así (**2 puntos**)

```
$ ./matrixGen 3
1 7 6
8 6 9
9 6 7
```

(**II**) Escribir un guión shell denominado *2matrixBully* que genere en salida estándar el elemento más grande de dos matrices cuadradas. *2matrixBully* toma la primera matriz (M1) de un fichero que toma como parámetro y la segunda (M2) de entrada estándar. Se invocaría, por ejemplo, como

```
$ ./matrixBully matrix1 < matrix2
```

1) Escribir una función denominada `matrixDimensions` que aplicada a un fichero que contiene una matriz MxN proporcione en la variable `nrR` su número de filas y en la variable `nrC` su número de columnas. (**2,5 puntos**)

2) Escribir una función denominada `matrixAccum` que aplicada a un fichero que contiene una matriz MxN proporcione en la variable `accummulated` la suma de todos sus elementos. (**1,5 puntos**)

3) Comprobar que el parámetro del guión *2matrixBully* es un fichero que existe y que tiene el mismo número de filas que de columnas. Escribir este número en la variable `N`.

4) Almacenar M2 en un fichero temporal *AuxM_2* y comprobar que este tiene el mismo número de filas que columnas y que este es igual a `N`

5) Escribir en salida estándar el elemento más grande de los acumulados de M1 y M2. (**1 punto**)

(**III**) Comprobar que funciona correctamente el mandato (**3 puntos**)

```
$ ./matrixGen 8 > ./matrix1 && ./matrixGen 8 | ./2matrixBully matrix1; rm matrix1
```

Tiempo: 1 hora y 20 minutos

Solución

matrixGen

```
#! /bin/bash
if [ $# -ne 1 ]; then
  echo Usage: matrixGen Dimension 1>&2
  exit 1
fi

for (( i = 0; i < $1; i++ )); do
  for (( j = 0; j < $1; j++ )); do
    echo -n "$(($RANDOM % 10)) "
  done
  echo
done
```

2matrixBully

```
#! /bin/bash

# (II.1) Escribir una función denominada matrixDimensions que aplicada a un fichero
#        que contiene una matriz MxN proporcione en la variable nrR su número de filas
#        y en la variable nrC su número de columnas.
function matrixDimensions ()
{
  # Contar columnas de M2
  #
  read fila_1 < $1
  nrC=0
  for elem in $fila_1
  do
    nrC=`expr $nrC + 1`
  done

  # Contar filas de M2
  #
  nrR=0
  while read fila; do
    nrR=`expr $nrR + 1`
  done < $1
  return
}

# (II.2) Escribir una función denominada matrixAccum que aplicada a un fichero
#        que contiene una matriz MxN proporcione en la variable accummulated
#        la suma de todos sus elementos.
function matrixAccum ()
{
  matrixDimensions $1
  accummulated=0
  while read fila
  do
    for number in $fila
    do
     accummulated=`expr $accummulated + $number`
    done
  done < $1
  return
}


# (II.3) Comprobar que el parámetro del guión 2matrixBully es un fichero que existe
#        y que tiene el mismo número de filas que de columnas.
#        Escribir este número en la variable N
if [ $# -ne 1 ]; then
  echo Usage: 2matrixBully M_1 1>&2
  exit 1
fi

# Contar filas y columnas de M1
matrixDimensions $1
columnCounter=$nrC
rowCounter=$nrR
if [ $columnCounter -ne $rowCounter ]; then
  echo Bad M1 file
  exit 1
fi
N=$columnCounter
```

```
#
# (II.4) Almacenar M2 en un fichero temporal m2 y comprobar que este tiene
#        el mismo número de filas que columnas y que este es igual a N
# Leo AuxM_2 desde la entrada estandar
echo -n > AuxM_2
for (( i = 0; i < $N; i++ )); do
  read fila
  echo $fila >> AuxM_2
done

# Contar filas y columnas de AuxM_2
matrixDimensions AuxM_2
columnCounter=$nrC
rowCounter=$nrR
if [ $columnCounter -ne $rowCounter ]; then
  echo Bad M_2 file. It has $columnCounter columns and $rowCounter rows
  exit 1
fi
if [ $columnCounter -ne $N ]; then
  echo Bad M_2 file. It must have $N rows and $N columns
  exit 1
fi

#
# (II.5) Escribir en salida estándar el elemento más grande de los acumulados de M1 y M2
matrixAccum $1
accum_M1=$accummulated
matrixAccum AuxM_2
accum_M2=$accummulated
if [ $accum_M1 -ge $accum_M2 ]; then
  echo $accum_M1
else
  echo $accum_M2
fi
rm AuxM_2
```