

Realizar un script denominado *mProductor*. Tiene un parámetro entero, entre 1 y 8. La invocación será pues, por ejemplo

```
$ ./mProductor 4
```

Realizar un script denominado *mConsumidor*. Tiene un parámetro entero, entre 1 y 8 y el nombre de un fichero. La invocación será pues, por ejemplo

```
$ ./mConsumidor 4 medias
```

La especificación del script *mProductor* es la siguiente:

1. Escribirá en *error estándar* el nombre y apellidos del autor.
2. Control del número de parámetros de entrada. El programa avisa y termina si el parámetro es incorrecto. **(1 punto)**
3. Generará por salida estándar una matriz cuadrada. Si el parámetro es P , la matriz será de $P \times P$ números aleatorios. El valor de cada número varía entre 0 y $P-1$. Tras el último número de cada fila, el programa producirá un fin de línea. Se espera pues una salida como la siguiente **(3 puntos)**

```
$ ./mProductor 4
0 1 0 2
2 3 1 1
0 2 0 2
3 1 2 3
```

La especificación del script *mConsumidor* es la siguiente:

4. Escribirá en *error estándar* el nombre y apellidos del autor.
5. Control del número de parámetros de entrada. El programa avisa y termina si el primer parámetro es incorrecto.
6. Leerá de *entrada estándar* P líneas, donde P es el primer parámetro del script. Cada línea está compuesta por un número indeterminado de números. Para cada línea se calculará su valor medio y se escribirá en el fichero parámetro, uno tras otro y separados por blancos (observar que el fichero parámetro acabará formado por una línea de P números). **(3 puntos)**
7. Ordenará por orden alfabético el fichero parámetro, manteniendo el mismo en una única línea. **(1 puntos)**

Comprobación: Ejecución en pipe

8. Debe ejecutar correctamente el siguiente mandato, de modo que no se nos pida nada por teclado **(2 puntos)**

```
$ ./mProductor 8 | ./mConsumidor 8
```

Lo que necesita saber para el punto 7

El siguiente bucle lee e imprime en pantalla, una a una, todas las líneas del fichero *file*:

```
while read line
do
    echo $line
done < file
```

Tiempo: 1 hora y 15 minutos

Entrega: En dos pasos;

1. Los ficheros *mProductor* y *mConsumidor* se comprimirán con el mandato

```
$ tar cvzf scripts.tar.gz ./mProductor ./mConsumidor
```

2. El fichero *scripts.tar.gz* se entregará en el campus virtual en la tarea correspondiente al grupo.

mProductor

```
#!/bin/bash

#
# 1. Escribirá en pantalla el nombre y apellidos del autor.
#
echo Juan Carlos Díaz Martín 1>&2

#
# 2. Control del número de parámetros de entrada. El programa avisa y termina si hay parámetros.
#
if [ ! $# -eq 1 ]; then
    echo Use: $(basename $0) P
    exit
fi

#
# 3. Generará por salida estándar una matriz cuadrada. Si el parámetro es P, la matriz será de
#     PxP números aleatorios
#
P=$1
echo Genero una matrix de $P x $P 1>&2
for (( i = 1 ; i <= $P; i++ ))
do
    for (( j = 1 ; j <= $P; j++ ))
    do
        palabra=`expr $RANDOM % $P`
        # echo $palabra
        echo -n "$palabra "
    done
    echo
done
```

mConsumidor

```
#!/bin/bash

#
# 4. Escribirá en error estándar el nombre y apellidos del autor.
#
echo Juan Carlos Díaz Martín 1>&2

#
# 5. Control del número de parámetros de entrada.
#     El programa avisa y termina si el primer parámetro es incorrecto.
#
if [ ! $# -eq 2 ]; then
    echo Use: $(basename $0) [1..8] file
    exit
fi

if [ $1 -lt 1 -o $1 -gt 8 ]; then
    echo Use: $(basename $0) [1..8] file
    exit
fi

#
# 6. Leerá de entrada estándar P líneas, donde P es el primer parámetro del script.
#
rm $2 2>/dev/null
P=$1
#echo Tomo una matrix de $P filas
for (( i = 1 ; i <= $P; i++ ))
do
    read linea
    #echo $linea
    acum=0
    for n in $linea
    do
        acum=`expr $n + $acum`
    done
    acum=`expr $acum / $P`
    echo -n "$acum " >> $2
done
echo >> $2
cat $2

#
# 7. Ordenará por orden alfabético el fichero parámetro, manteniendo el mismo en una única línea
#
rm aux1 2>/dev/null
read linea < $2
```

```
#echo he leído la línea $línea
for n in $línea
do
    #echo pongo en aux1 el número $n
    echo $n >> aux1
done
#echo aux1 es:
#cat aux1
#echo
#echo El fichero ordenado es:
sort < aux1 > sorted
#cat sorted

rm $2
while read num
do
    echo -n "$num " >> $2
done < sorted
echo >> $2

echo El fichero ordenado de una línea es:
cat $2
```