Física Computacional Tarea 1

Granados David

Marlon Brenes

1. Parte 1

1.1. Parte 1.a

Este sería el código a utilizar con sus explicaciones:

```
#! /bin/bash

i=$1 #Asigna el primer argumento dado al invocar el script a la 'variable' i
N=$2 #Asigna el segundo argumento dado al invocar el script a la 'variable' N

#Generar  los N archivos con los nombres adecuados
for ((j=0; j<i; j++)); do
  filename="rand${j}_${N}.dat"

  #Corre el archivo random.sh y asigna los n meros al archivo nuevo
  for ((k=0; k<N; k++)); do
    #Extrae los n meros y los asigna dentro de 'filename'
    number=$(bash random.sh | awk '/^[0-9]+$/{print;exit}')
  echo -e "$number" >> $filename
  done #Continua con la lectura del archivo

  #Los escribe en formato columna
  echo "Archivo $filename generado con $N n meros aleatorios."
done #Continua con la lectura del archivo ( ltima  l nea)
```

Code 1. 1.a.

1.2. Parte 1.b

Este sería el código a utilizar con sus explicaciones:

```
#! /bin/bash

paste rand*.dat > numbers.sh #Encuentra todos los archivos que cumplan con el ran*.dat y guarda en un archivo aparte los n meros asociados a este.
```

Code 2. 1.b.

1.3. Parte 1.c

Se utilizaría la función sort, al utilizar la -r antes de la definición del tipo de archivo cambia el orden.

```
sort -k 1 -n rand*.dat #Orden acendente

sort -k 1 -r rand*.dat #Orden descendente
```

Code 3. 1.c.

Esto 'sortearía' los archivos por sus características.

1.4. Parte 1.d

Se haría de la siguiente forma (cambiando la k por m o g se llega a las megas y gigas respectivamente):

```
du -k archivo
```

Code 4. 1.d.

El comando du en Bash es un comando que muestra el espacio en disco que ocupan archivos y directorios, su diferenciación se da con los '-*' donde en vez del asterisco se utilizan las letras para diferenciar el tamaño.

**Todas los caracteres que faltan son vocales tildadas.

2. Parte 2

2.1. Parte 2.a

Esto se hace utilizando los 'grep's para buscar dentro del archivo coincidencias entre los tres factores dados adjuntándolos como una sola condición con los |.

```
grep '^Brian,' usuarios.csv | grep ',.*\.com,' | grep ',Boeing$'
```

Code 5. 2.a.

Cuando se cumplen las condiciones se ven reflejadas en la terminal.

2.2. Parte 2.b

Se puede leer tal que: Saque los que en la primera columna cumplan con los nombres James o Paul dentro de usuarios.csv, además donde se cumpla que alguno de estos tenga los caractéres J o S dentro de *last_name* y además trabaje en Ad Astra.

```
grep -E '^(James|Paul),' usuarios.csv | grep -E ',[JS][^,]*,' | grep ',Ad Astra$'
```

Code 6. 2.b.

2.3. Parte 2.c

Se determina utilizando -c que cuenta todos los campos que cumplan con la condición de que M sea su primer caracter en el primer campo, para eso se utiliza el 'símbolo' de potencia.

```
grep -c '^M' usuarios.csv > aux.sh
```

Code 7. 2.c.

Se reescribe todo en un archivo llamado aux.sh

2.4. Parte 2.d

Se utilizó un comando llamado sep que ayuda a procesar el texto y hacer manipulaciones con estos. Luego estas manipulaciones hechas se guardan en el archivo original.

```
sed '1s/$/,phone_number/; 2,$s/$/,5555-5555/' usuarios.csv > usuarios.csv
```

Code 8. 2.d.

**Todas los caracteres que faltan son vocales tildadas.
