

Ejercicio 1 (33 puntos)

a)

- Logearse con su usuario al servidor “*ic.fisica.edu.uy*” vía el protocolo *ssh*. Crear una carpeta “*examen-<usuario>*”, donde el usuario entre corchetes, es su nombre de usuario en el servidor. Cambiarse a la carpeta de trabajo creada.

Sol:

```
ssh usuario@ic.fisica.edu.uy
```

- Crear un archivo llamado “*archivo.txt*” con el comando *touch*. Setear los permisos necesarios para que el dueño del archivo, puede leerlo, editarlo y ejecutarlo (en caso de que sea un ejecutable).

Sol:

```
touch archivo.txt
chmod u+rxw archivo.txt
```

- Renombrar el archivo creado “*archivo.txt*” con el nuevo nombre “*archivoNuevo.txt*”.

Sol:

```
mv archivo.txt archivoNuevo.txt
```

b)

- Copiar a su carpeta de trabajo el archivo “*covid1.txt*” que se encuentra en la ruta “*/home/introcomp-files/archivos-evaluaciones/archivos-2021*” del servidor *ic*.

Sol:

```
cp /home/introcomp-files/archivos-evaluaciones/archivos-2021/covid1.txt .
```

- Bajar del servidor ftp público “*ic.fisica.edu.uy*”, el archivo “*covid2.txt*”, que se encuentra en la ruta *archivos-evaluaciones/archivos-2021* relativa al servidor *ftp*.

Sol:

```
wget ftp://ic.fisica.edu.uy/home/introcomp-files/archivos-evaluaciones/archivos-2021/covid2.txt
```

- Concatenar los dos archivos de forma que el resultado quede en un archivo llamado “*covid.txt*”.

Sol:

```
cat covid1.txt covid2.txt > covid.txt
```

- Mostrar en pantalla solo la primera línea de encabezados del archivo “*covid.txt*”.

Sol:

```
tail -1 covid.txt
```

c)

- Desplegar en pantalla el mensaje: “La cantidad de líneas del archivo covid.txt es: <nr. Líneas>”, donde entre corchetes debe quedar sustituido la cantidad de líneas actual del archivo.

Sol:

```
echo "La cantidad de líneas del archivo covid.txt es: `wc -l covid.txt`"
```

- Mostrar en pantalla solo los datos de vacunación del departamento de Flores.

Sol:

```
grep Flores covid.txt
```

- Guardar en un archivo llamado “segundasDosis.txt”, los valores de las segundas dosis del archivo.

Sol:

```
awk '{ print $3}' covid.txt > segundasDosis.txt
```

- Desplegar en pantalla los departamentos cuyos porcentajes de segundas dosis supere el 60%.

Sol:

```
awk '$3>60 { print $1}' covid.txt
```

=====

d)

Escribir un script de Linux, que muestre el total de dosis aplicadas por departamento (dosis1 + dosis2), del archivo covid.txt

Sol:

```
#!/bin/bash
```

```
# recorro el archivo desde la linea 2 a la final (la 7)
```

```
for ((i=2; i<=7 ; i=i+1 ))  
do
```

```
    # para c/linea i recupero los campos de las dosis y los sumo  
    departamento=`head -$i ./covid.txt | tail -1 | awk '{ print $1 }'`\  
    dosisUno=`head -$i ./covid.txt | tail -1 | awk '{ print $2 }'`\  
    dosisDos=`head -$i ./covid.txt | tail -1 | awk '{ print $3 }'`\  
    dosisTotal=$((dosisUno + dosisDos))
```

```
    echo "La dosis total del dpto: $departamento es: $dosisTotal"  
done
```

```
## EOF
```

```
////////////////////////////////////
```

Ejercicio 2 (33 puntos)

program montecarlo

real, parameter :: PI = 3.141592

integer :: N = 10000

```

integer :: nrPtosInteriores
real    :: xa,ya,areaRectangulo
real    :: a = 2 ; b = 0.5

areaRectangulo = a*b
nrPuntosInteriores = 0

do i = 1,N
  ! --- genero el to aleatorio (xa,ya) en [0,1]*[0.1] y escaleo a [0,a]*[0,b]
  call random_number(xa)
  call random_number(ya)
  xa = xa*a
  ya = ya*b

  ! --- evaluo si el punto es interior a la elipse
  if ( (xa**2/a**2 + ya**2/b**2) < 1) then
    nrPuntosInteriores = nrPuntosinteriores + 1
  end if
end do

print*, "El area del cuadrante positivo de la elipse es: PI*a*b/4: ",PI*areaRectangulo/4
print *, "el area aprox. del cuadrante positivo de la elipse es: ",(areaRectangulo*nrPuntosInteriores)/N

end program montecarlo

```

////////////////////////////////////

Ejercicio 3 (34 puntos)

```

program maxOcurrencesCharacter

integer ,parameter      :: sizeArray=7
character, dimension(sizeArray) :: charArray = ('a','b','c','a','a','b','f')
character, dimension(sizearray) :: list      ! array con los caracteres nuevos q' voy hallando
character                :: element        ! character que recorre a charArray
integer , dimension(sizeArray) :: ocurrencias ! array con las ocurrencias de list
integer                  :: ultimo,pos,posFinded ! ultimo marca la posicion siguiente al ultimo
character en list
integer                  :: max,indexMax    ! inicialmente es 1, posFinded es la posicion donde
encuentro un
                                ! caracter en charArray, antes de c/busqueda es cero

ultimo = 1
pos = 1

do while (pos <= sizeArray)
  element = charArray(pos)
  call findElementInList(element,list,ultimo,posFinded)
  if (posFinded /= 0) then

```

```

    ocurrencias(posFinded) = ocurrencias(posFinded) + 1
else
    list(ultimo) = element
    ocurrencias(ultimo) = 1
    ultimo = ultimo + 1
end if
pos = pos + 1
end do

```

!----- Encuentro el maximo de ocurrences -----

!----- busco cual carcacter tiene mayor cant de ocurrencias -----

```

pos = 1
max = 0
indexMax = 1
do while (pos < ultimo)
    if (ocurrencias(pos) > max) then
        max = ocurrencias(pos)
        indexMax = pos
    end if
    pos = pos + 1
end do

```

```

print *, "El caracter con mayor ocurrencia es: ",list(indexMax)
print *, "La cantidad de ocurrencias del caracter: ",ocurrencias(indexMax)

```

!----- Subs -----

contains

subroutine findElementInList(element,list,ultimo,posFinded)

```

integer ,intent(in) :: ultimo
integer ,intent(out) :: posFinded
character,intent(in) :: element
character, dimension(:),intent(in) :: list
integer :: position

```

```

posFinded = 0
position = 1

```

!--- posFinded es 0 inicialmente, y si no se encuentra en el loop sigue en cero

```

do while ((position < ultimo).and.(posFinded == 0))
    if (element == list(position)) then
        posFinded = position
    else
        position = position + 1
    end if
end do

```

end do

end subroutine findElementInList

end program maxOcurrancesCharacter