

Ejercicio 1

a)

```
ftp introcomp-server.fisica.edu.uy
user: anonymous (sin clave)
get musica1.txt
get musica2.txt
quit
```

b)

```
wc -l musica1.txt
```

c)

```
ls -l musica1.txt
```

d)

```
cat musica1.txt musica2.txt > musica.txt
```

e)

```
echo "La cantidad de temas interpretado por Moby es: `grep Moby musica.txt | wc -l`"
```

Ejercicio 2

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

```
lines=`wc -l musica.txt| cut -d " " -f 1`
```

```
for ((i=2;i<=$lines;i=i+1))
```

```
do
```

```
  musico=`head -$i musica.txt| tail -1|cut -d " " -f 1`
```

```
  linea=`head -$i musica.txt|tail -1`
```

```
  if [ $musico == Moby ]
```

```
  then
```

```
    echo "$linea v" >> musicacool.txt
```

```
  else
```

```
    echo "$linea" >> musicacool.txt
```

```
  fi
```

```
done
```

```
#eof
```

Ejercicio 3

program sumaTernaria

```
character,dimension(3) :: numero1=['B','B','A']
character,dimension(3) :: numero2=['B','0','A']
character,dimension(4) :: suma=['0','0','0','0']
character :: digito1, digito2,digitoSuma,digitoCarry,digitoCarryNuevo,digitoCarryAux
integer :: imax
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!/ El primer acarreo es cero, asumo que el numero1 es //
!/ largo en caso de serlo. Luego recorro hasta el //
!/ largo de numero1 y si alcanzo el largo max del //
!/ numero2 antes, sus digitos son automaticamente //
!/ cero //
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
digitoCarry = '0'
imax=size(numero1)
```

```
do i = 1,imax
  digito1 = numero1(i)
  if (i > size(numero2)) then
    digito2 = '0'
  else
    digito2 = numero2(i)
  end if
  call sumarDigitos(digito1,digito2,digitoCarry,digitoSuma,digitoCarryNuevo)
  suma(i) = digitoSuma
  digitoCarry = digitoCarryNuevo
end do
```

```
!-----contemplo el máximo digito de suma si hay acarreo -----
if (digitoCarry /= '0') suma(imax+1)= 'A'
```

```
!----- El primer indice del array es el primer digito, pero los imprimo invertidos para su lectura normal -----
print*, "num1, num2 ", numero1(size(numero1):1:-1), " ", numero2(size(numero2):1:-1)
print*, "suma ", suma(size(suma):1:-1)
```

```
!=====
```

CONTAINS

subroutine sumarDigitos(d1,d2,dcarry,ds,dcarryNuevo)

```
character,intent(in) :: d1,d2,dcarry
character,intent(out) :: ds,dcarryNuevo
integer :: d1dec,d2dec,dcarryDec,sumaDec,digitoSumaDec
integer, parameter :: base=3
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!/ convierto los digitos a decimal, sumo y //
!/ si supera la base hay carry A y la suma //
!/ de los digitos es el digito corresp. al //
!/ resto de dividir la suma decimal por la //
!/ base //
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
if (d1=='0') d1dec=0
if (d1=='A') d1dec=1
if (d1=='B') d1dec=2
if (d2=='0') d2dec=0
if (d2=='A') d2dec=1
if (d2=='B') d2dec=2
if (dcarry=='0') dcarryDec=0
if (dcarry=='A') dcarryDec=1
```

```
sumaDec = d1dec + d2dec + dcarryDec
```

```
if (sumaDec > 2) then
  digitoSumaDec = mod(sumaDec,base)
  if (digitoSumaDec == 0) ds = '0'
  if (digitoSumaDec == 1) ds = 'A'
  if (digitoSumaDec == 2) ds = 'B'
  dcarryNuevo = 'A'
else
  if (sumaDec == 0) ds = '0'
  if (sumaDec == 1) ds = 'A'
  if (sumaDec == 2) ds = 'B'
  dcarryNuevo = '0'
end if
```

end subroutine sumarDigitos

end program sumaTernaria

Ejercicio 4

```
program maximo
```

```
real, parameter :: pi=3.14159265  
real :: max, xmax, ymax  
real :: deltax, deltax, xcord, ycord, fun  
integer :: N
```

```
deltax = 0.1  
deltay = 0.1  
N = floor(pi/deltax)
```

```
max = 0.0  
xmax = 0.0  
ymax = 0.0
```

```
do i = 1,N  
  do j = 1,N  
    xcord = i*deltax  
    ycord = j*deltay  
    fun = sin(xcord+ycord)  
    if (fun > max) then  
      max = fun  
      xmax = xcord  
      ymax = ycord  
    end if  
  end do  
end do
```

```
print *, "xm,ym,max: ",xmax,ymax,max
```

```
end program maximo
```