

Evaluación1

Eduardo Hndz
Lic. En Física
Universidad De Sonora
Progración y Lenguaje Fortran

October 30, 2017

1 Sphere

Se nos otorgó un código el cual servía para calcular el área de un cilindro, del cual se pedía, modificarlo para posteriormente calcular el área y volumen de una esfera. Tuve que probar el código con distintos datos para ver cómo funciona y al momento de querer calcular el V y el A de la esfera tuve problemas con el arreglo acerca del número de dígitos que podría aceptar el valor del área obtenido. El código quedó de la siguiente manera:

```
program sphere

! Calculate the surface area of a cylinder.
!
! Declare variables and constants.
! constants=pi
! variables=radius squared

implicit none      ! Require all variables to be explicitly declared

integer :: ierr
character(1) :: yn
real :: radius, area , volume
real, parameter :: pi = 3.141592653589793

interactive_loop: do

!   Prompt the user for radius and height
!   and read them.

write (*,*) 'Enter radius.'
read (*,*,iostat=ierr) radius
```

```

!   If radius and height could not be read from input,
!   then cycle through the loop.

      if (ierr /= 0) then
        write(*,*) 'Error, invalid input.'
        cycle interactive_loop
      end if

!   Compute area.  The ** means "raise to a power."

      area = 4*pi*radius*radius

      volume=(4./3.)*(pi*radius*radius*radius)

!   Write the input variables (radius, height)
!   and output (area) to the screen.

      write (*,'(1x,a7,f20.6,5x,a7,f20.6,5x,a7,f20.6)') &
        'radius=',radius,'area=',area, 'volume=', volume

      yn = ' '
      yn_loop: do
        write(*,*) 'Perform another calculation? y[n]'
        read(*,'(a1)') yn
        if (yn=='y' .or. yn=='Y') exit yn_loop
        if (yn=='n' .or. yn=='N' .or. yn==' ') exit interactive_loop
      end do yn_loop

    end do interactive_loop

end program sphere

```

2 Medias y Sumatorias

Aquí se nos pidió calcular la Media Harmónica, la Media Aritmética y la sumatoria de numeros dados por el usuario otorgador un código en el cual el problema que tuve fue que combinaba distintas variables, reales y enteras juntas, lo cual crasheaba mi programa. el código quedó como sigue:

```

program summation
implicit none
integer :: a
real:: sum, n,am,hm,m
print*, "This program performs summations. Enter 0 to stop."

```

```

open(unit=10, file="SumData.DAT")
!dividir enteros entre enteros
!nunca combinar números
sum = 0.0
n=0.0
m=0.0
do
  print*, "Add:"
  read*, a
  if (a == 0) then
    exit
  else
    sum = sum + a
    n=(n+1)
    m=m+1.0/a
  end if
  am=sum/n
  hm=n/m
  write(10,*) a
end do

  print*, "Summation =", sum
  print*, "Arithmetic mean =", am
  print*, "Harmonic mean=", hm
write(10,*) "Summation =", sum
write(10,*) "Arithmetic mean=", am
write(10,*) "Harmonic mean=", am

close(10)
end!

```

Los datos obtenidos se exportaban a un documento de texto el cual quedó de la siguiente manera:

```

      2
      3
      4
      5
      6
      7
      8
      9
     10
Summation =   55.0000000
Arithmetic mean=   5.50000000
Harmonic mean=   5.50000000

```

3 Leibniz

En esta actividad se nos pidió calcular el valor de PI con uso de la Serie de Leibniz con los valores de 10,20,30,40 y 50, el programa quedó como sigue:

```
Program Leibniz
  implicit none
!declaramos variables
integer::i,n !n será el valor máximo al que se quiere calcular
real::S,pi

print*, "dame al valor de n, donde n es el número de términos máximos"
read*, n

S=0
do i=0,n
  S=S+((-1)**i)/(2*float(i)+1))
end do

pi=S*4

print*, 'n=',n, 'pi=',pi

end program Leibniz
```

Los resultados obtenidos son los siguientes:

Table 1: aproximación de pi

n	Pi
10	3.23231578
20	3.18918467
30	3.17384195
40	3.16597891
50	3.16119814