

# Iniciando con Fortran

Jesús Valenzuela Nieblas

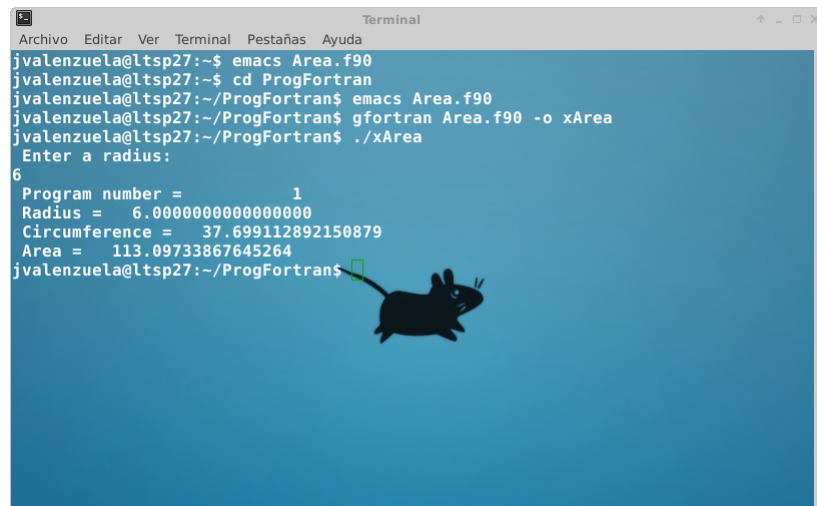
## 1. Introducción

En esta actividad, comenzamos a utilizar Fortran para la creación de pequeños programas que resuelvan ciertos problemas matemáticos

## 2. Programas en fortran

### 2.1. Área del círculo

Se nos pide crear un programa que calcule el área del círculo ingresando como dato únicamente el valor del radio.



```
Terminal
Archivo Editar Ver Terminal Pestañas Ayuda
jvalenzuela@ltsp27:~$ emacs Area.f90
jvalenzuela@ltsp27:~$ cd ProgFortran
jvalenzuela@ltsp27:~/ProgFortran$ emacs Area.f90
jvalenzuela@ltsp27:~/ProgFortran$ gfortran Area.f90 -o xArea
jvalenzuela@ltsp27:~/ProgFortran$ ./xArea
Enter a radius:
6
Program number =      1
Radius =  6.0000000000000000
Circumference =  37.699112892150879
Area =  113.09733867645264
jvalenzuela@ltsp27:~/ProgFortran$
```

! Area . f90 : Calculates the area of a circle, sample program

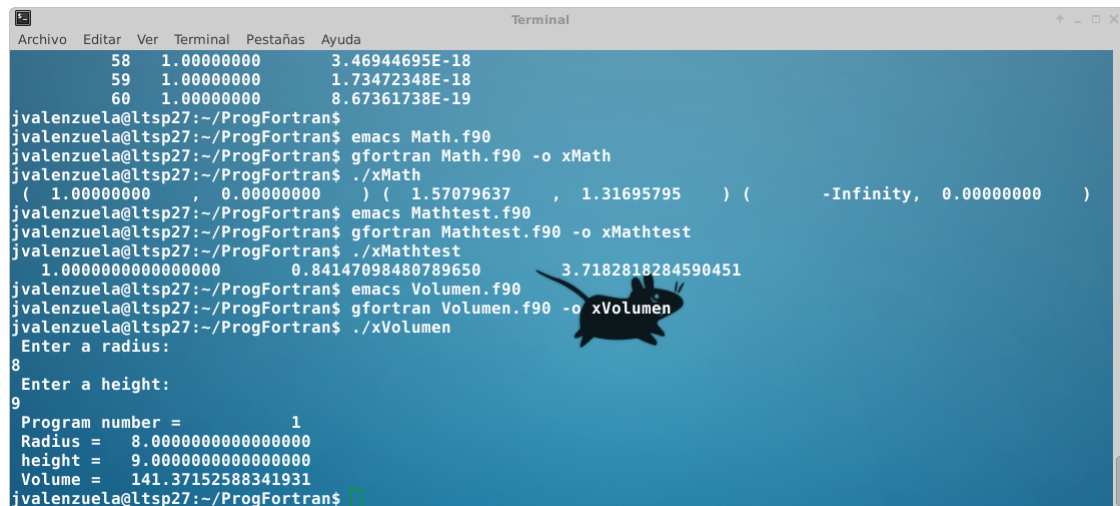
```

Program areacirculo ! Begin main program
Implicit None ! Declare all variables
Real *8 :: radius , circum , area ! Declare Reals
Real *8 :: PI = 4.0 * atan(1.0) ! Declare , assign Real
Integer :: model_n = 1 ! Declare , assign Ints
print * , 'Enter a radius:' ! Talk to user
read * , radius ! Read into radius
circum = 2.00 * PI * radius ! Calc circumference
area = radius * radius * PI ! Calc area
print * , 'Program number =' , model_n ! Print program number
print * , 'Radius =' , radius ! Print radius
print * , 'Circumference =' , circum ! Print circumference
print * , 'Area =' , area ! Print area
End Program areacirculo ! End main program code

```

## 2.2. Volumen de la esfera

En esta ocasión, se modificó el programa anterior para que calcul el volumen de una esfera ingresando únicamente su radio.



```

Terminal
Archivo Editar Ver Terminal Pestañas Ayuda
58 1.00000000 3.46944695E-18
59 1.00000000 1.73472348E-18
60 1.00000000 8.67361738E-19
jvalenzuela@ltsp27:~/ProgFortran$
jvalenzuela@ltsp27:~/ProgFortran$ emacs Math.f90
jvalenzuela@ltsp27:~/ProgFortran$ gfortran Math.f90 -o xMath
jvalenzuela@ltsp27:~/ProgFortran$ ./xMath
( 1.00000000 , 0.00000000 ) ( 1.57079637 , 1.31695795 ) ( -Infinity, 0.00000000 )
jvalenzuela@ltsp27:~/ProgFortran$ emacs Mathtest.f90
jvalenzuela@ltsp27:~/ProgFortran$ gfortran Mathtest.f90 -o xMathtest
jvalenzuela@ltsp27:~/ProgFortran$ ./xMathtest
1.0000000000000000 0.84147098480789650 3.7182818284590451
jvalenzuela@ltsp27:~/ProgFortran$ emacs Volumen.f90
jvalenzuela@ltsp27:~/ProgFortran$ gfortran Volumen.f90 -o xVolumen
jvalenzuela@ltsp27:~/ProgFortran$ ./xVolumen
Enter a radius:
8
Enter a height:
9
Program number = 1
Radius = 8.000000000000000
height = 9.000000000000000
Volume = 141.37152588341931
jvalenzuela@ltsp27:~/ProgFortran$

```

! Volumen . f90 : Calculates the area of a circle, sample program  
Program volumen ! Begin main program

```

Implicit None ! Declare all variables
Real *8 :: radius , height , volume , newradius ! Declare Reals
Real *8 :: PI = 4.0 * atan(1.0) ! Declare , assign Real
Integer :: model_n = 1 ! Declare , assign Ints
print * , 'Enter a radius:' ! Talk to user
read * , radius ! Read into radius
print * , 'Enter a height:' ! Talk to user
read * , height ! Pide la altura
newradius = 3 * radius - height
volume = 0.333333 * PI * height * newradius
print * , 'Program number =' , model_n ! Print program number
print * , 'Radius =' , radius ! Print radius
print * , 'height =' , height
print * , 'Volume =' , volume ! Print area
End Program volumen ! End main program code

```

## 2.3. Presición de la maquina

Se pide modificar el código dado para crear un programa que mida la precisión de la maquina.

```

!Limits.f90 Determines machine precision
Program Limits
  Implicit None
  Integer :: i , n
  Real *8 :: epsilon_m , one
  n=60 ! Establece el número de iteaciones
  ! Set initial values:
  epsilon_m = 1.0
  one = 1.0
  ! Within a DO-LOOP, calculate each step and print .
  ! This loop will execute 60 times in a row as i is
  ! incremented from 1 to n (since n = 60) :
  do i = 1, n , 1 ! Begin the do-loop
    epsilon_m = epsilon_m / 2.0 ! Reduce epsilon m
    one = 1.0 + epsilon_m ! Re-calculate one
    print *, i , one , epsilon_m ! Print values so far
  end do ! End loop when i>n

```

```

Terminal
Archivo Editar Ver Terminal Pestañas Ayuda

Enter a radius:
6
Program number = 1
Radius = 6.000000000000000
Circumference = 37.699112892150879
Area = 113.09733867645264
jvalenzuela@ltsp27:~/ProgFortran$ emacs Limits.f90
jvalenzuela@ltsp27:~/ProgFortran$ gfortran Limits.f90 -o xLimits
jvalenzuela@ltsp27:~/ProgFortran$ ./xLimits
1 1.500000000000000 0.500000000000000
2 1.250000000000000 0.250000000000000
3 1.125000000000000 0.125000000000000
4 1.062500000000000 6.250000000000000E-002
5 1.031250000000000 3.125000000000000E-002
6 1.015625000000000 1.562500000000000E-002
7 1.007812500000000 7.812500000000000E-003
8 1.003906250000000 3.906250000000000E-003
9 1.001953125000000 1.953125000000000E-003
10 1.000976562500000 9.765625000000000E-004
11 1.000488281250000 4.882812500000000E-004
12 1.000244140625000 2.441406250000000E-004
13 1.000122070312500 1.220703125000000E-004
14 1.000061035156250 6.103515625000000E-005
15 1.000030517578125 3.051757812500000E-005
16 1.000015258789062 1.525878906250000E-005
17 1.000007629394531 7.629394531250000E-006
18 1.000003814697265 3.814697265625000E-006
19 1.000001907348632 1.9073486328125000E-006
20 1.000000953674316 9.5367431640625000E-007
21 1.000000476837158 4.7683715820312500E-007
22 1.000000238418579 2.3841857910156250E-007
23 1.000000119209289 1.1920928955078125E-007
24 1.000000059604644 5.9604644775390625E-008
25 1.000000029802322 2.9802322387695312E-008
26 1.000000014901161 1.4901161193847656E-008
27 1.000000007450580 7.4505805969238281E-009
28 1.000000003725290 3.7252902984619141E-009
29 1.000000001862645 1.8626451492309570E-009
30 1.000000000931322 9.3132257461547852E-010
31 1.000000000465661 4.6566128730773926E-010
32 1.000000000232830 2.3283064365386963E-010
33 1.000000000116415 1.1641532182693481E-010
34 1.000000000058207 5.8207660913467407E-011
35 1.000000000029103 2.9103830456733704E-011
36 1.000000000014551 1.4551915228366852E-011
37 1.000000000007276 7.2759576141834259E-012
38 1.000000000003638 3.6379788070917130E-012
39 1.000000000001819 1.8189894035458565E-012
40 1.000000000000909 9.0949470177292824E-013
41 1.000000000000454 4.5474735088646412E-013

```

End Program Limits

## 2.4. Real

Se nos pide modificar el programa anterior para realizar las operaciones en precisión sencilla.

```

!Limitsencillo.f90 Determines machine precision
Program Limitsencillo
  Implicit None
  Integer :: i , n

```

```

Terminal
Archivo Editar Ver Terminal Pestañas Ayuda
60 1.00000000 8.67361738E-19
jvalenzuela@ltsp27:~/ProgFortran$ emacs Limitsencillo.f90
jvalenzuela@ltsp27:~/ProgFortran$ gfortran Limitsencillo.f90 -o xLimitsencillo
jvalenzuela@ltsp27:~/ProgFortran$ ./xLimitsencillo
1 1.50000000 0.50000000
2 1.25000000 0.25000000
3 1.12500000 0.12500000
4 1.06250000 6.25000000E-02
5 1.03125000 3.12500000E-02
6 1.01562500 1.56250000E-02
7 1.00781250 7.81250000E-03
8 1.00390625 3.90625000E-03
9 1.00195312 1.95312500E-03
10 1.00097656 9.76562500E-04
11 1.00048828 4.88281250E-04
12 1.00024414 2.44140625E-04
13 1.00012207 1.22070312E-04
14 1.00006104 6.10351562E-05
15 1.00003052 3.05175781E-05
16 1.00001526 1.52587891E-05
17 1.00000763 7.62939453E-06
18 1.00000381 3.81469727E-06
19 1.00000191 1.90734863E-06
20 1.00000095 9.53674316E-07
21 1.00000048 4.76837158E-07
22 1.00000024 2.38418579E-07
23 1.00000012 1.19209290E-07
24 1.00000000 5.96046448E-08
25 1.00000000 2.98023224E-08
26 1.00000000 1.49011612E-08
27 1.00000000 7.45058060E-09
28 1.00000000 3.72529030E-09
29 1.00000000 1.86264515E-09
30 1.00000000 9.31322575E-10
31 1.00000000 4.65661287E-10
32 1.00000000 2.32830644E-10
33 1.00000000 1.16415322E-10
34 1.00000000 5.82076609E-11
35 1.00000000 2.91038305E-11
36 1.00000000 1.45519152E-11
37 1.00000000 7.27595761E-12
38 1.00000000 3.63797881E-12
39 1.00000000 1.81898940E-12
40 1.00000000 9.09494702E-13
41 1.00000000 4.54747351E-13
42 1.00000000 2.27373675E-13
43 1.00000000 1.13686838E-13
44 1.00000000 5.68434189E-14
45 1.00000000 2.84217094E-14
46 1.00000000 1.42108547E-14

```



```

Real *4 :: epsilon_m , one
n=60 ! Establece el número de iteaciones
! Set initial values:
epsilon_m = 1.0
one = 1.0
! Within a DO-LOOP, calculate each step and print .
! This loop will execute 60 times in a row as i is
! incremented from 1 to n (since n = 60) :
do i = 1, n , 1 ! Begin the do-loop
    epsilon_m = epsilon_m / 2.0 ! Reduce epsilon m

```

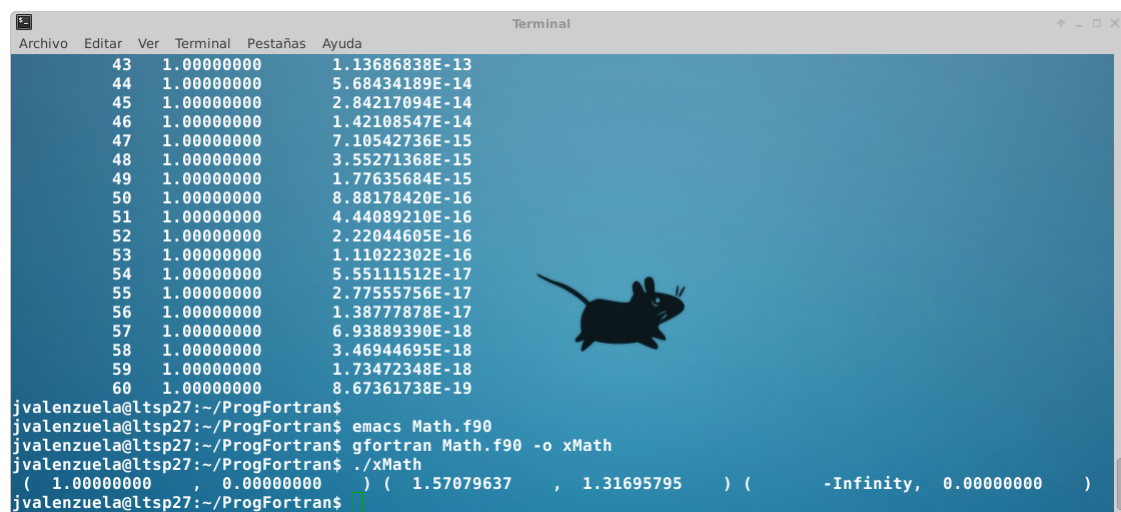
```

    one = 1.0 + epsilon_m ! Re-calculate one
    print *, i , one , epsilon_m ! Print values so far
end do ! End loop when i>n
End Program Limitsencillo

```

## 2.5. Funciones trigonométricas

Fortran maneja funciones trigonométricas. Aquí un ejemplo de ello.



```

Terminal
Archivo Editar Ver Terminal Pestañas Ayuda
43 1.00000000 1.13686838E-13
44 1.00000000 5.68434189E-14
45 1.00000000 2.84217094E-14
46 1.00000000 1.42108547E-14
47 1.00000000 7.10542736E-15
48 1.00000000 3.55271368E-15
49 1.00000000 1.77635684E-15
50 1.00000000 8.88178420E-16
51 1.00000000 4.44089210E-16
52 1.00000000 2.22044605E-16
53 1.00000000 1.11022302E-16
54 1.00000000 5.55111512E-17
55 1.00000000 2.77555756E-17
56 1.00000000 1.38777878E-17
57 1.00000000 6.93889390E-18
58 1.00000000 3.46944695E-18
59 1.00000000 1.73472348E-18
60 1.00000000 8.67361738E-19
jvalenzuela@ltsp27:~/ProgFortran$
jvalenzuela@ltsp27:~/ProgFortran$ emacs Math.f90
jvalenzuela@ltsp27:~/ProgFortran$ gfortran Math.f90 -o xMath
jvalenzuela@ltsp27:~/ProgFortran$ ./xMath
( 1.00000000 , 0.00000000 ) ( 1.57079637 , 1.31695795 ) ( -Infinity, 0.00000000 )
jvalenzuela@ltsp27:~/ProgFortran$

```

```

! Math.f90 : demostración de algunas funciones de fortran
Program Math !Comienza el programa principal
Complex *8 :: x= 1.0 , y=2 , z=0 !Declara las variabnles x, y , z
x = sqrt (x) ! función raíz cuadrada (square root)
y= asin (y) ! Llama a la función arcoseno
z= log (z) ! Función de exponencial
print * , x, y, z ! Print x,y,z
End Program Math

```

## 2.6. Math modificado

Se nos pide modificar el programa anterior para calcular la raíz cuadrada de -1, el arcoseno de 2 y el logaritmo de 0.

```

Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
47  1.00000000  7.10542736E-15
48  1.00000000  3.55271368E-15
49  1.00000000  1.77635684E-15
50  1.00000000  8.88178420E-16
51  1.00000000  4.44089210E-16
52  1.00000000  2.22044605E-16
53  1.00000000  1.11022302E-16
54  1.00000000  5.55111512E-17
55  1.00000000  2.77555756E-17
56  1.00000000  1.38777878E-17
57  1.00000000  6.93889390E-18
58  1.00000000  3.46944695E-18
59  1.00000000  1.73472348E-18
60  1.00000000  8.67361738E-19
jvalenzuela@ltsp27:~/ProgFortran$
jvalenzuela@ltsp27:~/ProgFortran$ emacs Math.f90
jvalenzuela@ltsp27:~/ProgFortran$ gfortran Math.f90 -o xMath
jvalenzuela@ltsp27:~/ProgFortran$ ./xMath
( 1.00000000 , 0.00000000 ) ( 1.57079637 , 1.31695795 ) ( -Infinity, 0.00000000 )
jvalenzuela@ltsp27:~/ProgFortran$ emacs Mathtest.f90
jvalenzuela@ltsp27:~/ProgFortran$ gfortran Mathtest.f90 -o xMathtest
jvalenzuela@ltsp27:~/ProgFortran$ ./xMathtest
1.0000000000000000 0.84147098480789650 3.7182818284590451
jvalenzuela@ltsp27:~/ProgFortran$

```

Program Math\_test

```

real *8 :: x=1 , y, z
y = sin (x)
z = exp (x) + 1.00
print * , x, y, z
End Program Math_test

```

## 2.7. Función

Se nos pide crear un programa que calcula el valor de una función  $f(x, y) = 1 + \sin(x y)$  definida por el usuario

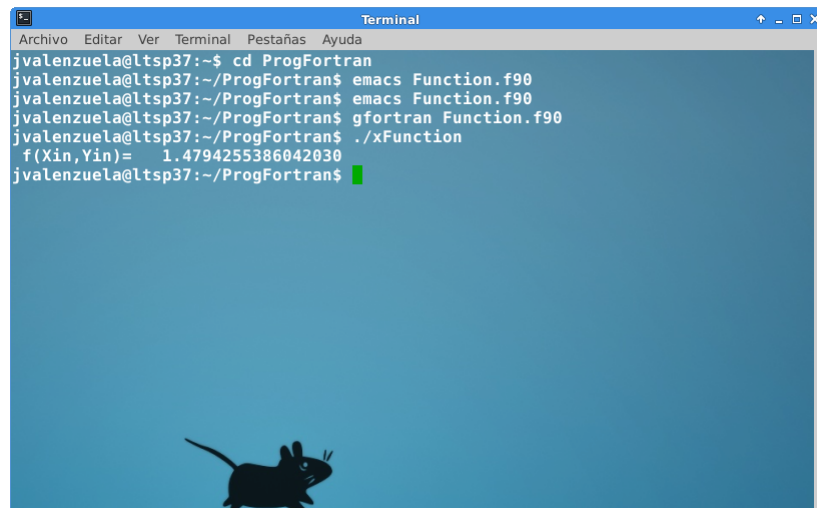
```

! Function . f90
Real *8 Function f (x,y)
  Implicit none
  Real *8 :: x, y
  f = 1.00 + sin (x*y)
End Function f
!

Program Main
  Implicit none
  Real*8 :: Xin =0.25, Yin =2.0, c, f !
  c= f ( Xin, Yin )
  write ( *,* ) 'f(Xin,Yin)=', c

```



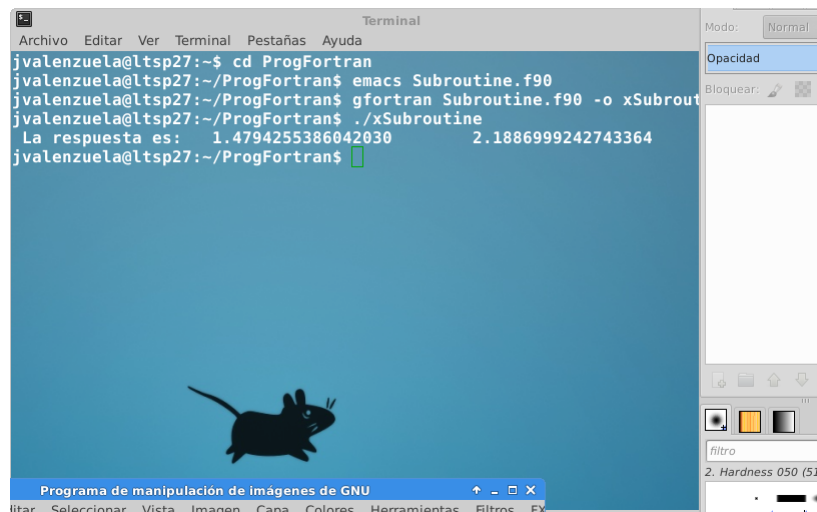
A terminal window titled "Terminal" with a menu bar (Archivo, Editar, Ver, Terminal, Pestañas, Ayuda) and a blue background. It shows a series of commands and their outputs. At the bottom, there is a small black silhouette of a mouse.

```
jvalenzuela@ltsp37:~$ cd ProgFortran
jvalenzuela@ltsp37:~/ProgFortran$ emacs Function.f90
jvalenzuela@ltsp37:~/ProgFortran$ gfortran Function.f90
jvalenzuela@ltsp37:~/ProgFortran$ ./xFunction
f(Xin,Yin)= 1.4794255386042030
jvalenzuela@ltsp37:~/ProgFortran$
```

End Program Main

## 2.8. Subrutina

Fortran además de funciones también maneja subrutinas. He aquí el ejemplo.

A terminal window titled "Terminal" with a menu bar (Archivo, Editar, Ver, Terminal, Pestañas, Ayuda) and a blue background. It shows commands to compile and run a Fortran subroutine. At the bottom, there is a small black silhouette of a mouse. To the right of the terminal, there is a sidebar with various settings like "Modo: Normal", "Opacidad", and "Bloquear".

```
jvalenzuela@ltsp27:~$ cd ProgFortran
jvalenzuela@ltsp27:~/ProgFortran$ emacs Subroutine.f90
jvalenzuela@ltsp27:~/ProgFortran$ gfortran Subroutine.f90 -o xSubroutine
jvalenzuela@ltsp27:~/ProgFortran$ ./xSubroutine
La respuesta es: 1.4794255386042030 2.1886999242743364
jvalenzuela@ltsp27:~/ProgFortran$
```

```
Subroutine g(x,y,ans1,ans2)
  Implicit None
```



```

    Real (8) :: x,y,ans1,ans2
    ans1 = sin (x*y) + 1
    ans2 = ans1**2
    End Subroutine g
!
Program Mainprogram
  Implicit None
  Real *8 :: Xin = 0.25, Yin = 2.00, Gout1, Gout2
  call g(Xin, Yin, Gout1, Gout2)
  write (*,*) 'La respuesta es:',Gout1, Gout2
End Program Mainprogram

```