

FORTRAN PROGRAMS

EXAMPLE 1 Convert a temperature from Centigrade to Fahrenheit.

! Program for Centigrade to Fahrenheit

Program centi_Fahren

Implicit none

Real :: C, F

Print *, 'Please Type the value of Temperature in Centigrade'

Read(*,*) C

F = C * 9.0/5.0 + 32.0

Write(*, "(1x,'Temperature in Fahrenheit is', f10.3)") F

End program centi_Fahren

EXAMPLE 2 To find the slope and Midpoint of a Line

! Program to find the slope and Midpoint of a line

Program slope_midpoint

Implicit none

Real :: x1, y1, x2, y2, slope, X, Y

Print *, 'Please Type the first point x1, y1'

Read(*,*) x1, y1

Print *, 'Now Please Type the second point x2, y2'

Slope = (y2 - y1) / (x2 - x1)

X = (x1 + x2) / 2.0

Y = (y1 + y2) / 2.0

Write(*, "(1x,'Slope is', f10.3,/, 'Midpoint is',1x,2(f10.3,1x))") Slope, X, Y

End program slope_midpoint

EXAMPLE 3 Area of a Triangle.

! Program for finding Area of a Triangle

Program Area_Triangle

Implicit none

Real :: A, B, C, S, Area

Print *, 'Please Type the sides of a Triangle A, B, C'

Read(*,*) A, B, C

S = (A + B + C) / 2.0

Area = SQRT(S * (S - A) * (S - B) * (S - C))

Write(*, "(1x,'Area of the Triangle is ', f10.3)") Area

End program Area_Triangle

EXAMPLE 4 Velocity and Distance of a Particle.

! Program to find velocity and distance of a particle

Program Vel_Distance

Implicit none

Real :: a, t, S, V

Print *, 'Please Type acceleration a and time t of a particle'

Read(*,*) a, t

S = (1.0/2.0) * a * t **2

V = a*t

Write(*, "(1x,'Time =',1x,f10.3,/, 'Distance =',1x,f10.3,/, 'Velocity is', f10.3)") t, S, V

End program Vel_Distance

EXAMPLE 5 Critical constant of Gas.

! Program for determining critical constant of Gas

Program critical_constants

Implicit none

Real :: a, b, R, Tc, Pc, Vc

R = 0.0821

Print *, 'Please Type a and b'

```

Read(*,*) a, b
Tc = 8.0 * A / (27.0 * R * b)
Pc = a / (27.0 * b ** 2)
Vc = 3.0 * b
Write(*, "(1x,'Vander Waals constants a and b for a gas are',1x,2 f10.7)") a, b
Write(*, "(1x, 'Critical Temperature Tc =',1x,f14.7,/, 'Critical Pressure Pc =',1x,f14.7 &
&,/, 'Critical Volume Vc =',1x,f14.7)") Tc, Pc, Vc
End program critical_constants

```

EXAMPLE 5 Sum series $1+x+x^2+x^3+\dots$

```

! Program for sum of series
Program sum_series
Implicit none
Real :: x, sum
Integer:: N
Print *, 'Please Type x and N'
Read(*,*) x, N
sum = 1.0
I = 1
reading: Do
sum = sum + x ** I
I = I + 1
If( I >= N) exit reading
Enddo reading
Write(*, "(1x,' Sum of the series =',1x, f10.3)") sum
End program sum_series

```

EXAMPLE 6 Quadratic Equation

! Program for solving Quadratic equation

Program quadratic_equation

Implicit none

Real :: a, b, c, D, RP, IMP, X, X1, X2

Integer:: FLG

Print *, 'Please Type a, b and c'

Read(*,*) a, b, c

D = b * b - 4.0 * a * c

If (D < 0) FLG = 1

If(D = 0) FLG = 2

If(D > 0) FLG = 3

Select case(FLG)

Case (1)

RP = - b / (2.0 * a)

IMP = SQRT(ABS (D)) / (2.0 * a)

Write(*, "(1x, 'The Roots are complex conjugates', //, 1x, 'Real Part =' , f10.3, //, &
& 'Imaginary Part =' , f10.3)") RP, IMP

Case(2)

X = - b / (2.0 * a)

Write(*, "(1x, 'Equal roots', //, 1x, 'Root =' , f10.3)") X

Case(3)

X1 = (- b + SQRT(D)) / (2.0 * a)

X2 = (- b - SQRT(D)) / (2.0 * a)

Write(*, "(1x, 'The Roots are real and distinct', //, 1x, 'First Root =' , f10.3, //, &
& 'Second Root =' , f10.3)") X1, X2

End select

Case default

Write(*, "(1x, 'Wrong data')")

End program quadratic_equation

EXAMPLE 7 Prime Number.

! Program to check whether Prime Number

Program Prime_Number

Implicit none

Integer :: N, I, IR

Print *, 'Please Type Number N'

Read(*,*) N

I =2

10 Continue

If(I > N/2)Then

Write(*, "(1x, 'Number' ,1x, I8, 'is Prime')") N

Go to 20

Endif

IR = N – N/I * I

If(IR == 0)Then

Write(*, "(1x, 'Number' ,1x, I8, 'is not Prime')") N

Elseif(IR /= 0) Then

I = I +1

Go to 10

Endif

20 Continue

End program Prime_Number

EXAMPLE 8 Fibonacci Numbers 0, 1, 1, 2, 3, 5, 8, 13,

! Program to print all the Fibonacci Numbers less than MAXN

Program Fibonacci_Numbers

Implicit none

Integer :: N0, N1, N, MAXN

Print *, 'Please Type Maximum Limit MAXN'

Read(*,*) MAXN

N0 = 0

N1 = 1

Write(*, "(1x, I5, 1x, I5)", advance='no') N0, N1

10 N = N1 + N0

If(N >= MAXN) Then

Write(*, "()", advance='yes')

Stop

Endif

Write(*, "(1x, I5)", advance='no') N

End program Fibonacci_Numbers

EXAMPLE 9 SINE X Series

! Program to print the value of SIN(X)

Program Sine_X

Implicit none

Integer :: N

Real :: x, sum

Print *, 'Please Type the value of x and Maximum Limit N (i.e. No. of Terms of Sine X & series)'

Read(*,*) x, N

Sum = x

I = 0

Doloop : Do

Sum = Sum + ((-1.0) I) * (x ** I / Fact(I))**

```

I = I + 2
If( I <= N) exit Doloop
Enddo Doloop
Write(*, "(1x, 'Sine X is', 1x, f15.8)" ) Sum
End program Sine_X

```

```

Function Fact( K )
Integer :: K, L
Real :: M, Fact
If( K = 0) Fact = 1.0
If ( K /= 0) then
    M = 1.0
    L = 1
    Doloop1: DO
        M = M * (M + 1.0)
        L = L + 1
        If( L >= K) exit Doloop1
    Enddo Doloop1
    Fact = M
Endif
End Function Fact

```

EXAMPLE 10 Searching for character.

! Program to verify whether a string contains a particular character

```
Program search_character
```

```
Implicit none
```

```
Character(len=1) :: FLG
```

```
Character(len=80)::string
```

```
Print *, 'Please Type a character you want to search in a string'
```

```
Read(*,*) FLG
```

```
Print *, 'Please Type a string'
```

```
Read(*, "(a80)" ) string(1: 80)
```

```

I = 0
Do J = 1, N
  IF(string(J:J) == 'FLG')Then
    I = I + 1
  Endif
Enddo
Write(*, "(1x, 'Character',1x, a1, 'appears', 1x, I5, 'times in a string')") FLG
End program search_character

```

EXAMPLE 11 Reading the name of the month and printing the first three characters.

! Program to read the name of a month and print only the first three characters

```

Program print_month
Implicit none
Character(len=15) :: month
Print *, 'Please Type a month'
Read(*,"(a15)") month
Write(*, "(1x, 'First three character of the Month is', 1x, a3)") month(: 3)
End program print_month

```

EXAMPLE 12 Reverse of a string.

! Program to find the reverse of a string

```

Program reverse_string
Character(len=80) :: string, Rev
Implicit none
Integer:: J, N
Print *, 'Please Type a length of a string N'
Read(*,*) N
Print *, 'Please Type a string'
Read(*,"(a)") string(1:N)
Rev(1:N) = ' '
Rev = string(1:1)
Do J = 2, N
  Rev = string(J:J) // Rev

```

Enddo

Write(*, "(1x, 'Reverse of a string', 1x, a, 'is', 1x, a)") string, Rev

End program reverse_string

EXAMPLE 13 Maximum and / or Minimum number of an array.

! Program to find the maximum and / or minimum number of an array

Program maxmin_array

Implicit none

Integer, allocatable, dimension(:) :: A

Integer:: I, J, Max, Min, ist, ist1

Open(11, file= 'INPUT.txt', status = 'old' , action = 'read')

Open(12, file= 'OUTPUT.txt', status = 'unknown', action = 'write')

I = 1

Swap: Do

Read(11, "(I5)", iostat = ist) A(I)

If (ist /= 0) exit Swap

I = I + 1

Enddo Swap

Allocate(A(I))

Rewind(11)

Reading: Do

Read(11, "(I5)", iostat = ist1) A(I)

If(ist1 /= 0) exit reading

Enddo reading

Max = A(1)

Min = A(1)

Do J = 1, I

If(A(J) >= Max) Max = A(J)

If(A(J) < Min) Min = A(J)

Enddo

Write(*, "(1x, 'Maximum number of an array is ', 1x, I5, /, 1x, 'Minimum number & of an array is', 1x, I5)') Max, Min

End program maxmin_array