

6.Go-To Statement:**6.1 Un-conditional Go To Statement****Example:** write program for print sequence number and its square

sol: i=1

10 i=i+1

j=i**2

write(*,30) i,j

go to 10

30 format(2x,/, 'i=',i4, 'j=',i8)

stop

end

6.2 Computed Go To Statement:

general form:

Go to (n1,n2,n3,.....),i**Example:** Write a program to compute and print(y) and, print" error code" for another value of K :

$$y = \sqrt{A + 2B^2 + 3C^3} \quad \text{at } K=1$$

$$Y = |A - 2B - 3C| \quad \text{at } K=2$$

$$Y = \ln\left(\frac{A}{B-C}\right) \quad \text{at } K=3$$

Sol:

Read *,A,B,C,D,K

GO TO (10,20,30),K

PRINT *,'ERROR CODE'

STOP

10 Y=SQRT(A+2*B**2+3*C**3)

GO TO 40

20 Y=ABS(A-2*B-3*C)

GO TO 40

```
30 Y=ALOG10(A/(B-C))
40 PRINT*, 'THE VALUE=', Y
STOP
END
```

7. If- Statements :

7.1-IF-STATEMENT arithmetic

If (a) n1,n2,n3

where n1:statement at (a) is negative

n2:statement at (a) is zero

n3:statement at (a) is positive

Example: Write a program to compute and print (y) from the equations. Use If- statement.

$$y = \begin{cases} x^2 & x < 0 \\ 1 & x = 0 \\ x+3 & x > 0 \end{cases}$$

solution:

```
      read *,x
      if(x) 10,20,30
10    y=x2
      go to 40
20    y=1
      go to 40
30    y=x+3
40    print*, 'y=', y
      stop
      end
```

Example: Write a program to compute and print (y) from the equations. Use If- statement.

$$z = \begin{cases} x+y & x > y \\ x-y & x = y \\ x*y & x < y \end{cases}$$

```

1  read *,x,y
   if (x-y) 10,20,30
10  z=x*y
   go to 40
20  z=x-y
   go to 40
30  z=x+y
40  print *, 'z=',z
   stop
   end

```

Example: Write a program to compute and print (y) from the equations. Use If- statement. Where(a=2)

$$y = \begin{cases} \cos^2(x) - x & x < -a \\ 1 - \frac{x}{a} & -a \leq x \leq a \\ \log\left(\frac{x}{a}\right) & x > a \end{cases}$$

```

c  this program for find log
   read*,x,a
   if(x.gt.-2)then
     y=cos(x)**2-x
   else
     if(-2.le.x.and.x.le.2) then
       y=.5*(1-x/a)
     if (x.gt.2) then
       y=log(x/2)
     endif
   endif
   print*, 'y=',y
   stop
   end

```

7.2 Logical If Statement:

An important part of any programming language are the *conditional statements*. The most common such statement in Fortran is the if statement, which actually has several forms. The simplest one is the logical if statement:

if (*logical expression*) *executable statement*

This has to be written on one line. This example finds the absolute value of x:

if (x .LT. 0)

x = -x

If more than one statement should be executed inside the if, then the following syntax should be used:

if (*logical expression*) then

statements

endif

The most general form of the if statement has the following form:

if (*logical expression*) then

statements

elseif (*logical expression*) then

statements

:

:

else

statements

end if

The execution flow is from top to bottom. The conditional expressions are evaluated in sequence until one is found to be true. Then the associated code is executed and the control jumps to the next statement after the end if.

Example: what is the result after run the following program:

y=0.0;x=5.5

10 if(x.LT.y) x=x+1.0

y=y+1.0

if(y.LE.6.0) go to 10

stop

end RUN :x=6.5 y=7.0

Example: write program for input names of (100 student) and the name of each one ,exam number and three marks ,print ,exam no. student name and the average?

solution:

```
Character B*10
I=1
20 read *,N,B,X1,X2,X3
   Ave=(X1+X2+X3)/3.0
   Write (*,10) N,B,Ave
10  format(2x,'exam no',I3,A10,'Average=',f6.2)
   I=I+1
   IF(I.LE.100) GO TO 20
stop
end
```

7-3:Nested if statements

if statements can be nested in several levels. To ensure readability, it is important to use proper indentation. Here is an example:

```
if (x .GT. 0) then
  if (x .GE. y) then
    write(*,*) 'x is positive and x = y'
  else
    write(*,*) 'x is positive but x < y'
  endif
elseif (x .LT. 0) then
  write(*,*) 'x is negative'
else
  write(*,*) 'x is zero'
endif
```

You should avoid nesting many levels of if statements since things get hard to follow.