

Programming Language: FORTRAN

Er. Shiva K. Shrestha (HoD)

Department of Computer Engineering,
Khwopa College of Engineering



FORTRAN - FORmula TRANslatiOn

ForTran is general purpose, imperative programming suited for numerical & scientific computation.

Features

- ▶ Simple Language
- ▶ M/c Independent
- ▶ Expresses Complex Math
- ▶ Efficient Execution
- ▶ Storage Allocation
- ▶ Freedom in Code Layout

Limitations

- ▶ ‘fixed form’ source format
- ▶ Lacks
 - ▶ Inherent Parallelism
 - ▶ Dynamic Storage
 - ▶ Numeric Portability
 - ▶ User-defined data structures
 - ▶ Explicit Recursion
- ▶ Reliance on Unsafe Storage & sequence Association Feature

Steps in Writing Program

1. Writing Source Code

1. Column 1: Blank, or "c" or "a" or "+" for comments
2. Column 2-5: Statement Label [Optional]
3. Column 6: Continuation of previous line [Optional]
4. Column 7-72: Statements
5. Column 73-80: For programmer's comment

2. Compilation
& Linking
3. Execution

	1	2-5	6	7-72	73-80
	c			Column 1: Blank, or a "c" or "*" for comments	
2.				write(*,*)'This is statement which is labeled as 900'	
	c		+	Column 6: Continuation of Previous Line (Optional)	
3.				Column 7-72: Actual Instructions or statements for execution	
	c		+	Column 73-80: Sequence Number(Optional, Rarely Used Today)	

Character Set & Escape Sequences

4

► Character Set

- ▶ Letters: (A to Z) or (a to z)
- ▶ Digits: (0 to 9)
- ▶ Symbols: + - / * . , ' _ \$ ()

► Escape Sequences

Escape Sequence	Purpose
\n	New Line (Line feed)
\b	Backspace
\t	Horizontal Tab
\r	Carriage Return
\f	Form feed
\'	Single quote
\"	Double quote
\\\	Backslash
\a	Alert
\?	Question mark

Sample Program

5

ForTran Program	C Program	Comment
real p,t,r,i	void main(){ float p,t,r,l;	Variable Declaration
WRITE(*,*)'Enter value of P: '	printf("Enter value of P: ");	Formatted Output Statement
read(*,*)p	scanf("%f",&p);	Formatted Input Statement
write(*,*)'Enter value of T: '	printf("Enter value of T: ");	Formatted Out
read(*,*)t	scanf("%f",&t);	Formatted In
WRITE(*,*)'Enter value of R: '	printf("Enter value of R: ");	Formatted Out
read(*,*)r	scanf("%f",&r);	Formatted In
i=(p*t*r)/100	i=(p*t*r)/100;	Operators, Operands & Expression
write(*,*)'The interest is',i	printf("The interest is %.2f.",i);	Formatted Out
END	}	End of Program

Er. Shiva K. Shrestha (HoD, Computer Department)
2019-02-17

Structure of ForTran Program

- a) Program Name
- b) Declaration Section
- c) Statements
- d) Stop [Optional]
- e) End

Program Name

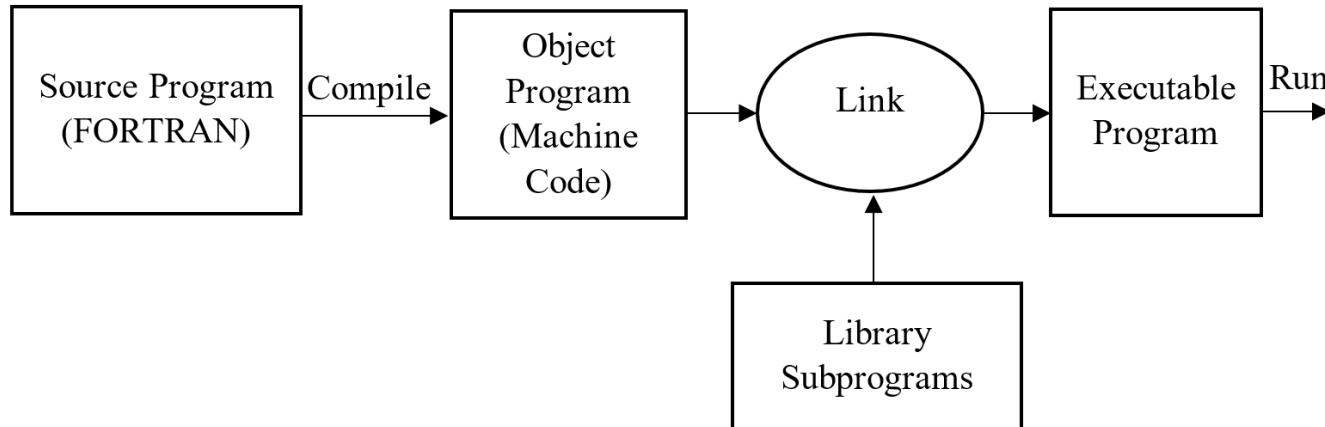
Declaration Section

Statements

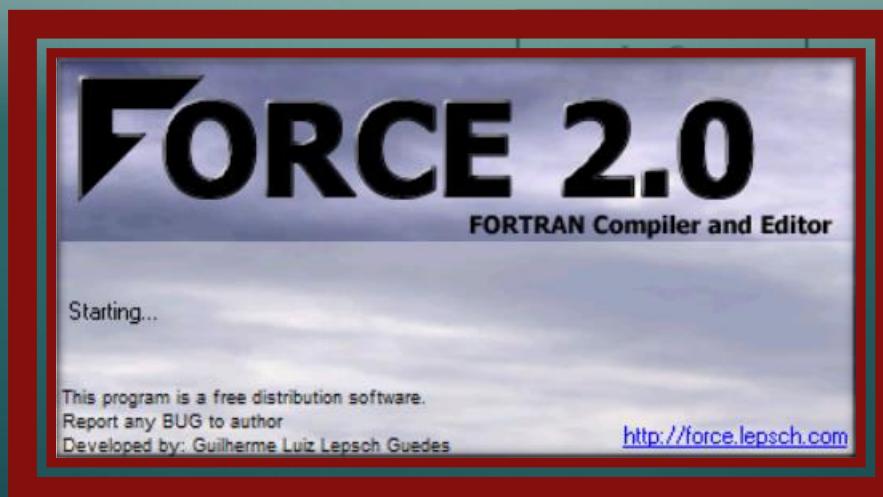
Stop [Optional]

End

Steps in Writing & Execution of a FORTRAN Program

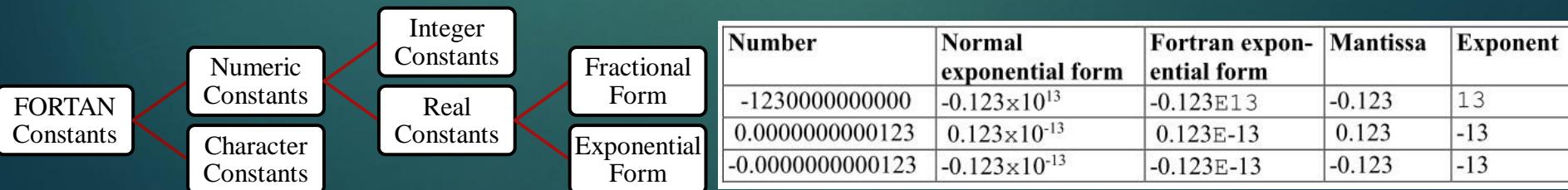


Er. Shiva K. Shrestha (HoD, Computer Department)
2019-02-17



Data Types, Variables & Constants

Data Type	Bytes	Description
INTEGER	4	Represents both positive & negative integral (non-fractional) numbers
REAL	4	Represents both positive & negative integral numbers with fractional parts
DOUBLE PRECISION	8	Same as REAL but using twice the storage space & greater precision.
COMPLEX	8	Represents an ordered pair of REAL data: real and imaginary components
LOGICAL	4	Represents the boolean data representing TRUE or FALSE
CHARACTER	1	Represents one alphanumeric character character.
CHARACTER*n	n	Represents multiple characters, where the maximum n is 32,767.



Calculate Area of a Circle

9

Er. Shiva K. Shrestha (HoD, Computer Department)
2019-02-17

```
1      program circle
2      real r, area
3 c      This program reads a real number r and prints
4 c      the area of a circle with radius r.
5      write(*,*) 'Enter Radius of Circle:'
6      read(*,*) r
7      area=3.14159*
8      +r*r
9      write(*,900)'Area=',area
10 900  format(A10,F10.2)
11      PAUSE
12      STOP
13      END
```

D:\[C]\Ch11\f_2.exe

Enter Radius of Circle:

5

Area= 78.54

PAUSE statement executed

To resume execution, type go. Other input will terminate the job.

E#4: Program reads three sets of **too small** and **too large** numbers and displays them to illustrate the concept of range of real constants.

```
1      real x,y !real type variable declaration like float x, y in C.
2      write(*,*)'Enter x & y:' !like printf("Enter x & y: "); in C.
3      read(*,*)x,y           !scanf("%f%f", &x,&y); in C.
4      write(*,*)'x:',x,' y:',y !like printf("x:%f y:%f", x, y); in C.
5      PAUSE
6      end
7      !end of a Fortran program
     !is a symbol used to comment in Fortran program like /* */ in C.
```

D:\[C]\Ch11\f_4.exe

```
Enter x & y:  
-786655553225586952233554458665.2258663211145589662235  
55522353655869944555222555555555.235588886665545885  
x: -7.86655587E+029 y: 5.55223535E+034
```

D:\[C]\Ch11\f_4.exe

```
Enter x & y:  
-3.4E38  
3.4E-38  
x: -3.3999995E+038 y: 3.39999989E-038
```

D:\[C]\Ch11\f_4.exe

```
Enter x & y:  
1234566789101112131415161718  
-325699845557899665555665223  
x: 1.23456679E+027 y: -3.25699845E+026
```

E#5: Program to read and display character constants of different length

```
1 character x*30,y,z*80 !character variable declaration.  
2 write(*,*) 'Enter name, gender & address:'  
3 read(*,*) x,y,z !reading three character variable  
4 write(*,*) 'Content of x:',x !printing content of x  
5 write(*,*) 'Content of y:',y !printing content of y  
6 write(*,*) 'Content of z:',z !printing content of z  
7 PAUSE  
8 end
```

Er. Shiva K. Shrestha (HoD)
2019-02-17

Variables

- ▶ data_type list of variables (separated by commas)
 - ▶ integer a
 - ▶ integer x, y, z
 - ▶ real l, b, h
 - ▶ character name*30, gender, address*50

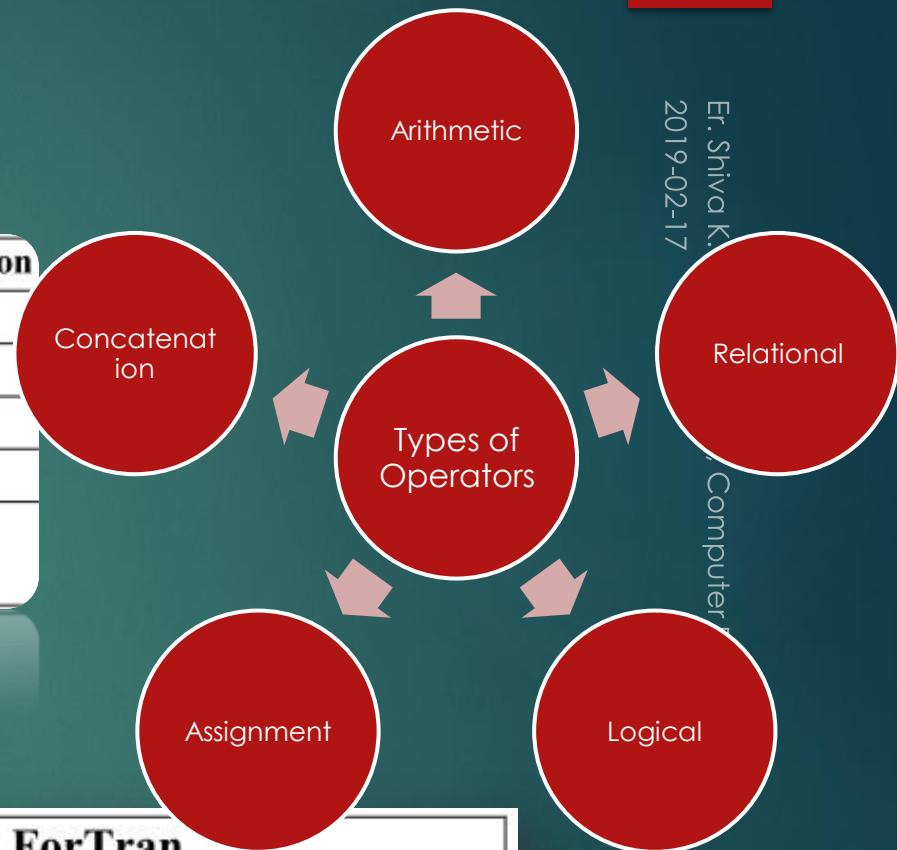
Implicit Type Declaration

Declaration	Comment
implicit integer(b)	Variables starting with b are considered as integer. Therefore, the letter those makes variables integer are b,i,j,k,l,m,n
implicit integer(p,u-z), real (m)	This declares that variable names starting with p,u,v,w,x,y,z or i,j,k,l,n are integers; m is real
implicit character *50(r-t)	It means variable name starting with r, s, t are character variables having capacity at most 50 characters.

Operators & Expressions

► Arithmetic Operators

Symbol	Operation	arithmetic expression
+ (plus)	addition	$x+y$
- (minus)	subtraction	$x-y$
* (asterisk)	multiplication	$x*y$
/ (slash)	division	x/y
(double asterisk)	exponentiation	$x^{}y$



In general mathematics	In ForTran
$2ax^4+bx^3+cx^2+2dx/5$	$2*a*x**4+b*x**3+c*x**2+(2*d*x/5)$
$A^{10/15}$	$A**\left(10.0/15.0\right)$
$a+b$	$(a+b)/(c+d)$
$c+d$	
$(a/b)^{p+30}$ and $(a/b)^{(p+30)}$	$(a/b)**p+30$ and $(a/b)**\left(p+30\right)$

Relational & Logical Operators

Relational operators in C	Equivalent symbol in ForTran
less than(<)	.lt.
less than or equal to (<=)	.le.
greater than(>)	.gt.
greater than or equal to (>=)	.ge.
equal to (==)	.eq.
equal to (!=)	.ne.
Logical operators in C	Equivalent symbol in ForTran
and (&&)	.and.
or ()	.or.
not (!)	.not.

for (i = 1 ; i < n ; i++)	.for(i = 1, i < n, i++).
if (condition)	.if(condition).
else	.else.

String Concatenation & Assignment Operators

- ▶ String Concatenation Operator
 - ▶ Two CHARACTER strings can be joined together in a single called concatenation operator. The concatenation operators is represented by a double forward slash //. It is also known as character operator.
- ▶ Assignment Operator (=)
 - ▶ Assignment operators are used to assign the result of an expression to a variable. The usual assignment operator is '='. The general form is
 - ▶ variable = expression

Library/Built-in/Intrinsic Functions

Function	Meaning	Argument type	Return type
abs(x)	x	Real or integer	Real or integer
exp(x)	e^x	Real	real
sqrt(x)	\sqrt{x}	Real	real
log(x)	$\log_e x$	Real ($x > 0$)	real
log10(x)	$\log_{10} x$	Real ($x > 0$)	real
sin(x)	sin x	Real, in radian measure	real
cos(x)	cos x	Real, in radian measure	real
tan(x)	tan x	Real, in radian measure	real
asin(x)	$\sin^{-1} x$	Real, $-1 \leq x \leq 1$	Real, in radian measure
acos(x)	$\cos^{-1} x$	Real, $-1 \leq x \leq 1$	Real, in radian measure
atan(x)	$\tan^{-1} x$	Real	Real, in radian measure
sinh(x)	sinh (x)	Real	real
cosh(x)	cosh (x)	Real	real
tanh(x)	tanh (x)	Real	real
int(x)	*	Real	integer
nint(x)	**	Real	integer
max(x1,x2,...)	**	Real or integer	Real or integer
	* converts the argument to an integer value		
	** rounds x to the nearest integer value		
	*** returns the maximum value of x1, x2, ...		

E#6: Using Library Functions

17

```
1      real x,y,z,m,r,ni      ! variable declaration
2      Write(*,*) 'Enter x, y & z:'
3      read(*,*) x,y,z
4      m=max(x,y,z)
5      r=int(y)
6      ni=nint(z)
7      write(*,*) 'max(x,y,z):',m,'\\n content of r=int(y):',r,
8      +'\\n content of ni=nint(z):',ni
9      write(*,*) 'absolute value of x:',abs(x)
10     write(*,*) 'Square root of y+z:',sqrt(y+z)
11     PAUSE
12     end
```

Er. Shiva K. Shrestha (HoD, Computer Department)
2019-02-17

D:\[C]\Ch11\f_6.exe

```
Enter x, y & z:
4.5
5.5
6.5
max(x,y,z): 6.5
content of r=int(y): 5.
content of ni=nint(z): 7.
absolute value of x: 4.5
Square root of y+z: 3.46410155
```

Task

Questions: Write a program in FORTRAN

- a) to find the volume of a room when height, width and length is given.
- b) to find the area and circumference of a circle when radius is given.
- c) to find surface area and volume of a sphere when radius of the sphere is given.
- d) to find value of $f(x)=x^2 +3$ if value of x is given.
- e) to convert the given Centigrade measure to Fahrenheit using relation $F=1.8C+32$.
- f) to compute equivalent resistance of two resistors R_1 and R_2 when they are connected in series and parallel connection.
- g) to read two end points of a line, compute their mid-point and display it.

Note: Run a ForTran program using Force2.0 compiler.

Formatted & Unformatted I/O Statements

- ▶ I/O Functions
 - ▶ `read(UNIT=unit_no, FMT=format_no)`
 - ▶ `write(UNIT=unit_no, FMT=format_no)`
- ▶ Format Statements
 - ▶ Label `FORMAT(format_specification1, format_specification2, ...)`

Common format code letters:

- i. I – integer
- ii. A – alphabet
- iii. E – real numbers, exponent notation
- iv. F – real numbers, fixed point format
- v. X – horizontal skip (space)
- vi. / - vertical skip (newline)
- vii. T – Tab
- viii. Quote – “”

I Format

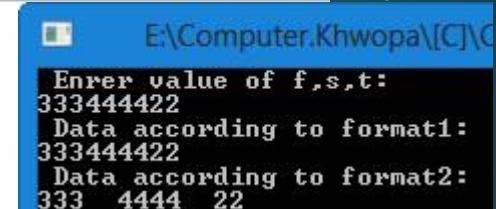
- ▶ I format is for integer. Its general form is `lw`. Where w is the width of the integer data.
- ▶ For example:
 - ▶ -13 has width 3,
 - ▶ 1245 has width 4 and
 - ▶ +5 has width 2.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
-	1	3	1	2	4	5	+	5											

```

1      integer f,s,t
2      write(*,*) 'Enrer value of f,s,t:'
3      read(*,1)f,s,t      !reading data in format 1
4 1    format(I3,I4,I2)   !format statement 1
5      write(*,*) 'Data according to format1:'
6      write(*,1)f,s,t    !writing or printing data in format 1
7      write(*,*) 'Data according to format2:'
8      write(*,2)f,s,t    !writing or printing data in format 2
9 2    format(i3,2x,I4,2x,I2) !format statement 2
10     PAUSE
11     end

```



A Format

- ▶ It is used for character data. The general form is Aw, where w is the width. The w is optional here.
- ▶ For example,

Character x*12

X= 'Mt. Everest'

- ▶ Output for **format(A)** and **format(A4)** are as illustrated in the following figure:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
M	t	.	E	V	e	r	e	s	t										

Figure: illustration of **format(A)** for printing data

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
M	t	.	E																

Figure: illustration of **format(A4)** for printing data

E Format

- ▶ Its general form is Ew.d, where w is the total width and d is the decimal width of the mantissa.
- ▶ For example:
 - ▶ -0.12E-12; w = 9, d = 2
 - ▶ 1.23E12; w = 7, d = 2

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
-	0	0	2	E		-	1	2	1	.	2	3	E	1	2				

Figure: illustration of format(E9.2,E7.2) for reading data

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
		-	0	0		2	E	-	1	2			1	.	2	3	E	1	2

Figure: illustration of format(1x, E9.2, 2x, E7.2) for printing data

Illustrates the concept of reading and writing real data using E format

```
1      real f,s
2      write(*,*)'Enter values of f and s:'
3      read(*,1)f,s
4 1    format(E9.2,E7.2)
5      write(*,*)'Values stored at f and s are:'
6      write(*,2)f,s
7 2    format(2x,E9.2,2x,E7.2)
8      pause
9      end
```

```
E:\Computer.Khwopa\[C]\C_Lecturer_Materials\C_Cpt11_FORTAN\f_9.exe
Enter values of f and s:
66666.333 7777.333
Values stored at f and s are:
 0.67E+05  .78E+04
PAUSE statement executed
To resume execution, type go. Other input will terminate the job.
```

F Format

- ▶ The general form of this format is Fw.d, where w is the total width and d is decimal width of the number.
- ▶ For example, -12.345 has total width(w) 7 and decimal width(d) 3.
- ▶ format(F7.3,F6.4) & format(F7.3,3x,F6.4) are shown below:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
-	1	2	.	3	4	5	1	.	2	3	4	6							

Figure: illustration of format(F7.3,F6.4) for reading data

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
-	1	2	.	3	4	5				1	.	2	3	4	6				

Figure: illustration of format(F7.3,3x,F6.4) for printing data

Illustrates the concept of reading and writing real data using F format

```
1      real f,s
2      write(*,*)'Enter values of f and s:'
3      read(*,1)f,s
4 1    format(F7.3,F6.4)
5      write(*,*)'Values stored at f and s are:'
6      write(*,2)f,s
7 2    format(F7.3,3x,F6.4)
8      pause
9      end
```

```
E:\Computer.Khwopa\[C]\C_Lecturer_Materials\C_Cpt11_FORTAN\f_8.exe
Enter values of f and s:
44.3333 3.44444
Values stored at f and s are:
44.333 3.4440
PAUSE statement executed
To resume execution, type go. Other input will terminate the job.
```

X Format

- x format is used to skip column(s) while printing data. Its general form is nX, where n is the number of columns to be skipped.
- If we want to skip two columns between –13 and 1245 & 1245 and +5, the format specification becomes format(I3,2X,I4,2X,I2). Then the output will be as:

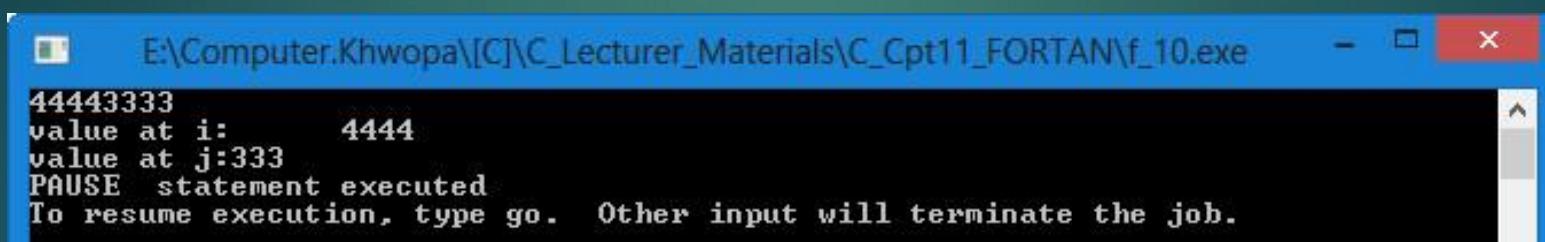
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
-	1	3			1	2	4	5			+ 5								

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
-	1	3			1	2	4	5			+ 5								

/ Format

- / (Slash) is used to skip to the next line like \n in C.

```
1      read(*,1)i,j !why I and j are not declared here?
2 1      format(I4,I3)
3      write(*,2)i,j
4 2      format('i:',I10/'j:',I3) !use n slash for skipping n lines
5      pause
6      end
```



The screenshot shows a Windows command-line interface window titled 'E:\Computer.Khwopa\[C]\C_Lecturer_Materials\C_Cpt11_FORTAN\f_10.exe'. The window displays the following output:

```
44443333
value at i:    4444
value at j:333
PAUSE statement executed
To resume execution, type go. Other input will terminate the job.
```

T Format & Quote Format

► T Format

- ▶ This format is used to give tab. Its general form is Tn. Where n stands for column from which the output must start. It is used only for output statements. For example, let us take a format statement, format(F8.2,T12,I3).

► Quote Format

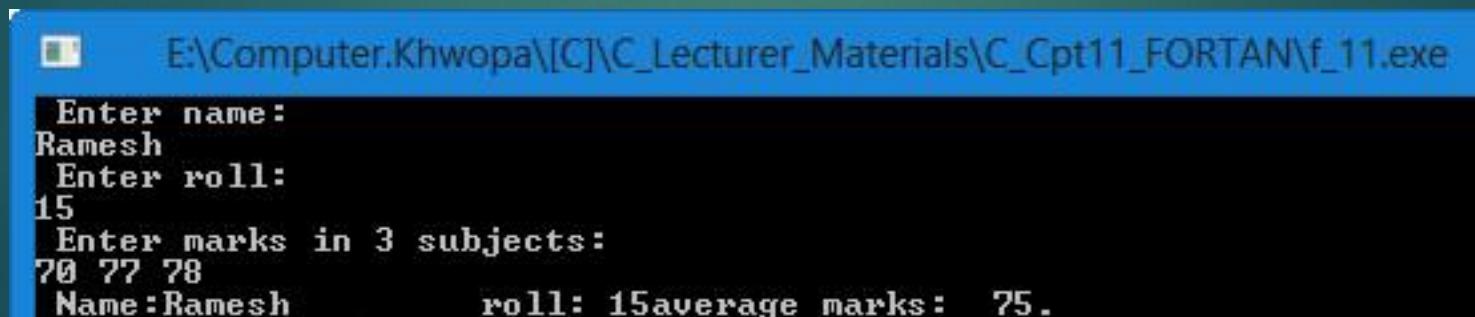
- ▶ This is similar to “ ” in C. whatever is in the single quote, will be printed as it is. For example, if we need to print NEPAL the format statement for this task is format('NEPAL').

Unformatted I/O

- ▶ read statement:
 - ▶ `read*`, `a1, a2, a3, ...` where `a1, a2, a3, ...` are the list of the variables.
 - ▶ When this statement is executed, the cursor blinks on the screen waiting for getting data. The supplied data must match with the type of the corresponding variable. The character data must be enclosed in single quotes. If there are more than one input data, they must be separated by spaces.
- ▶ print statement:
 - ▶ `write*`, `I1, I2, I3, ...` where `I1, I2, I3, ...` is the list of items to be printed.
 - ▶ The items may include variables, constants, arithmetic expressions and character constants. These will be printed continuously with blank spaces between two items.

Illustrates the concept of unformatted I/O using read & print statements.

```
1      character Name*15
2      integer roll
3      real m1,m2,m3
4      print*, 'Enter name:'
5      read*, name
6      print*, 'Enter roll:'
7      read*, roll
8      print*, 'Enter marks in 3 subjects:'
9      read*, m1,m2,m3
10     print*, 'Name:', name, 'roll:', roll, 'average:', (m1+m2+m3)/3
11     PAUSE
12     end
```



```
E:\Computer.Khwopa\[C]\C_Lecturer_Materials\C_Cpt11_FORTAN\f_11.exe
Enter name:
Ramesh
Enter roll:
15
Enter marks in 3 subjects:
70 77 78
Name:Ramesh      roll: 15average marks:  75.
```

Control Structures:

goto, Logical IF, Arithmetic IF, Do loops

- ▶ goto
 - ▶ Unconditional goto
 - ▶ Computed goto
- ▶ Logical if
- ▶ Arithmetic if
- ▶ Do Loops
- ▶ Implied Do

E#12: Unconditional goto

32

Backward Jump vs. Forward Pass

```
1      integer x
2  1    write(*,*)'Enter x :'
3    read(*,*) ,x
4    write(*,*) 'x=' ,x
5    goto 1
6    end
```

```
E:\Computer.Khwopa\[C]\C_Lecturer_Materials\FC\FC11\FC11_11.F90
Enter value of variable x :
3
The value stored at variable x is: 3
Enter value of variable x :
5
The value stored at variable x is: 5
Enter value of variable x :
```

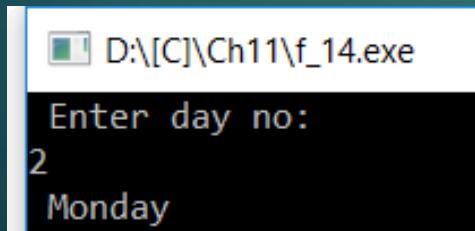
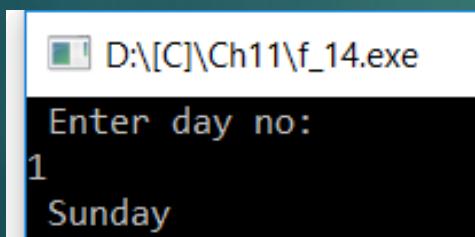
```
1      integer x
2      goto 20
3      write(*,*)'Enter x :'
4      read(*,*) x
5  20   write(*,*) 'x=' ,x
6      pause
7      End
```

```
E:\Computer.Khwopa\[C]\C_Lecturer_Materials\FC\FC11\FC11_11.F90
The value stored at variable x is: 8071304
PAUSE statement executed
To resume execution, type go. Other input will terminate
```

Computed goto

- ▶ `goto (k1, k2, k3, ..., kn), i`; where $k_1, k_2, k_3, \dots, k_n$ are statement numbers.

```
1      integer day;
2      write(*,*) 'Enter day no:'
3      read(*,*) day
4      goto(15,20), day
5      15    write(*,*) 'Sunday'
6      goto 23 ! goto stop statement.
7          ! otherwise control goes to immediate next statement.
8      20    write(*,*) 'Monday'
9      goto 23
10     23   stop
11     End
```



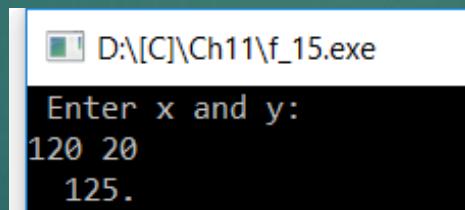
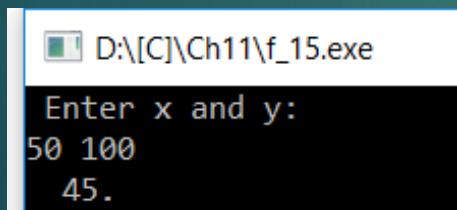
- ▶ Suppose: `goto(5,5,5,2,3,66), i`
 - ▶ 5 if the value of i is 1,2 or 3
 - ▶ 2 if the value of i is 4
 - ▶ 3 if the value of i is 5
 - ▶ 66 if the value of i is 6

Logical if

- ▶ For example if $x=100$ and $y=20$ then
- ▶ Is $x > y$? Yes. It means true
- ▶ Is $x = y$? No. It means false
- ▶ Is $(x > y) \text{ and } (x = y)$? \rightarrow true and false. It results in false.
- ▶ Is $(x > y) \text{ or } (x = y)$? \rightarrow true and false. It results in true.
- ▶ Is $\text{not} ((x > y) \text{ or } (x = y))$? \rightarrow not(true and false). \rightarrow not(true). Therefore, the final result becomes false.

E#15: WAP to read two real numbers x and y, add five to x if $x > y$, subtract five from it if $x \leq y$ and display the result.

```
1      real x, y
2      write(*,*) 'Enter x and y:'
3      read(*,*) x, y
4      if(x.gt.y) goto 1
5      x=x-5
6      goto 2
7 1    x=x+5
8 2    write(*,*) x
9      pause
10     end
```



E#16: WAP to evaluate the following expression.

$$f(x) = ax^2 + bx + 5 \text{ if } x < 5$$

$$f(x) = bx^2 + ax - 15 \text{ if } 5 \leq x$$

```

1   real a,b,x,fx
2   write(*,*) 'Enter a, b & x:'
3   read(*,*) a,b,x
4   if(x.lt.5) goto 1
5   goto 2
6   1   fx=a*x**2+b*x+5
7   write(*,*) 'The value of f(x):',fx
8   goto 10
9   2   fx=b*x**2+a*x-15
10  write(*,*) 'The value of f(x):',fx
11  goto 10
12  10  pause
13  end

```

D:\[C]\Ch11\f_16.exe

Enter a, b & x:

5 6 2

The value of f(x): 37.

D:\[C]\Ch11\f_16.exe

Enter a, b & x:

5 6 7

The value of f(x): 314.

E#17: WAP to check whether a positive integer read from the keyboard is palindrome or not. A number is palindrome if its reverse is equal to the number itself.

```

1  integer n,sum,r,x
2  sum=0
3  write(*,*) 'Enter a positive integer:'
4  read(*,*) n
5  x=n
6  2  if(x.eq.0)goto 1
7      r=mod(x,10)
8      sum=sum*10+r
9      x=x/10
10 goto 2
11 1  if(sum.eq.n)then
12      write(*,*) 'The number',n,'is a palindrome.'
13  else
14      write(*,*) 'The number',n,'is not a palindrome.'
15  endif
16  pause
17 end

```

```

D:\[C]\Ch11\f_17.exe
Enter a positive integer:
12321
The number 12321 is a palindrome.

D:\[C]\Ch11\f_17.exe
Enter a positive integer:
12345
The number 12345 is not a palindrome.

```

E#18: Evaluate $\sin(x) = x - x^3 / 3! + x^5 / 5! - x^7 / 7! \dots$

38

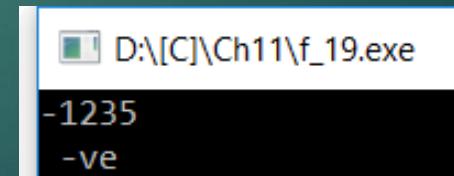
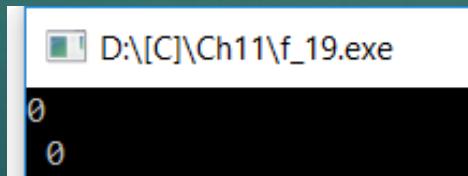
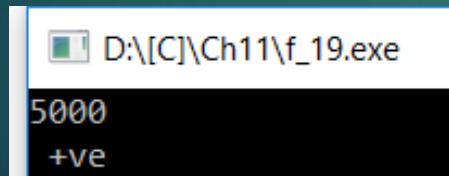
```
1      real deg, rad, sum, term, i
2      sum=0
3      i=0
4      write(*,*) 'Enter angle in degree:'
5      read(*,*) deg
6      rad=deg*3.14/180 !to convert degree to radian but it is not mandatory
7      term=rad
8      write(*,*) 'Term:',term !to print first term
9      3 if(abs(term).lt. 0.000001)goto 1
10     i=i+2
11     sum=sum+term
12     term=-term*rad*rad/(i*(i+1))
13     write(*,*) 'Term:',term !to print term
14     goto 3
15  1 write(*,*) 'sin(',deg,')=',sum !to print sum of terms
16     pause
17   end
```

```
E:\Computer.Khwopa\[C]\C_Lecturer_Materials\C_Cpt11_FORTAN\f_18.exe
Enter the value of angle in degree:
30
Term: 0.523333371
Term: -0.023888234
Term: 0.000327122863
Term: -2.13313569E-006
Term: 8.11414669E-009
sin( 30.)= 0.499770105
```

Arithmetic if

E#19: WAP to read an integer from the user and display appropriate message whether it is positive, zero or negative using the concept of arithmetic if statement.

```
1      real x
2      read(*,*)x
3      if(x)1,2,3
4      1      write(*,*) '-ve'
5      goto 100
6      2      write(*,*) '0'
7      goto 100
8      3      write(*,*) '+ve'
9      goto 100
10     100 pause
11     End
```



E#20: This program to calculate all roots of a quadratic equation. Here, if $b^2 - 4ac < 0$, control goes to statement 5, to 4 if $b^2 - 4ac = 0$ and to 10 if $b^2 - 4ac > 0$

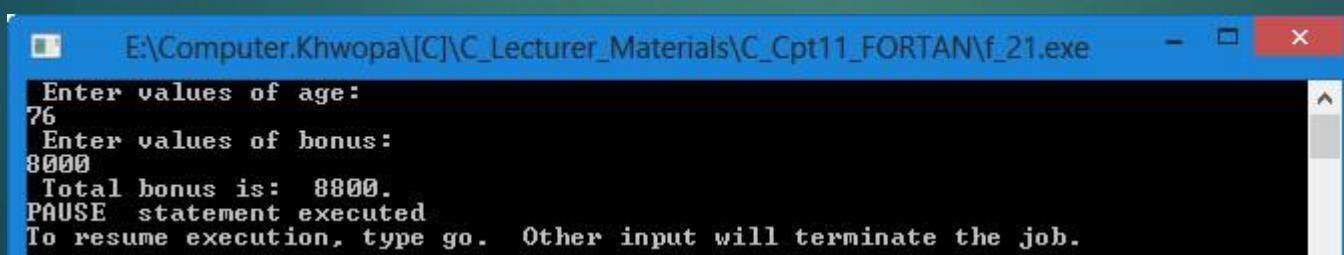
```

1  real a,b,c,r1,r2,d,rp,ip
2  write(*,*) 'Enter a, b & c:'
3  read(*,*) a,b,c
4  if(b*b-4*a*c) 5,4,10
5  5 rp=-b/2*a
6  ip=sqrt(abs(b*b-4*a*c))/2*a
7  write(*,*) 'The imaginary roots are: ',rp,'+i',ip,'and',rp,'-i',ip
8  goto 100 ! goto stop statement.
9  ! otherwise control goes to immediate next statement.
10 4 r1=-b/2*a
11 write(*,*) 'Roots are real and equal which are:',r1,r1
12 goto 100
13 10 r1=(-b+sqrt(b*b-4*a*c))/2*a
14 r2=(-b-sqrt(b*b-4*a*c))/2*a
15 write(*,*) 'Roots are real and unequal and which are:',r1,r2
16 100 pause
17 end

```

Simple if statement

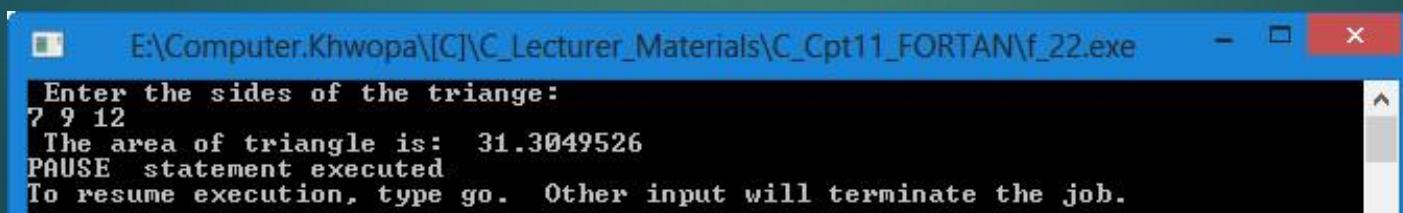
```
1      real age, bonus
2      write(*,*) 'Enter age:'
3      read(*,*),age
4      write(*,*) 'Enter bonus:'
5      read(*,*), bonus
6      if(age.gt.60) then
7          bonus=bonus+bonus*1.0/10.0
8      endif
9      write(*,*) 'Total bonus is:',bonus
10     pause
11     end
```



```
E:\Computer.Khwopa\[C]\C_Lecturer_Materials\C_Cpt11_FORTAN\f_21.exe
Enter values of age:
76
Enter values of bonus:
8000
Total bonus is: 8800.
PAUSE statement executed
To resume execution, type go. Other input will terminate the job.
```

E#22: A program to calculate the area of a triangle when the length of three sides is given. This program illustrates the concept of if...else, relational operators and logical operators.

```
1      real a,b,c,s,area
2      write(*,*) 'Enter sides of the triange:'
3      read(*,*),a,b,c
4      if((a+b) .gt. c .and. (b+c) .gt. a .and. (a+c) .gt.b) then
5          s=(a+b+c)/2
6          area=sqrt(s*(s-a)*(s-b)*(s-c))
7          write(*,*) 'The area of triangle is:',area
8      else
9          write(*,*) 'Invalid sides of the triangle.'
10     endif
11     pause
12 end
```



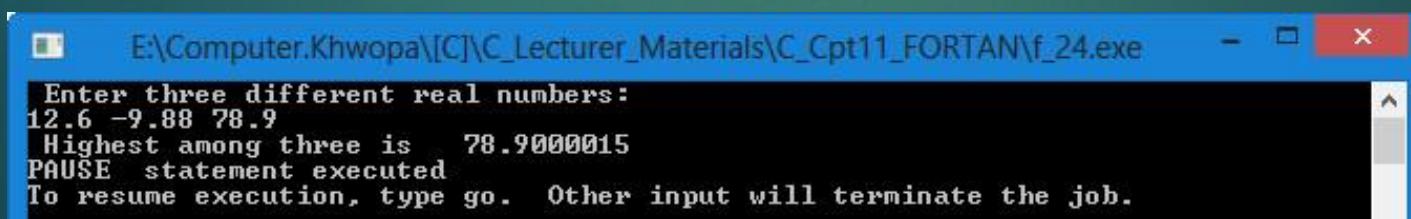
E#23: Write a program to check whether a year entered by a user is leap year or not. This example illustrates the concept of nested if...then...else statement.

```
1  integer year
2  write(*,*)'Enter a year:'
3  read(*,*) year
4  if(mod(year,4).eq.0)then
5      if(mod(year,100).eq.0)then
6          if(mod(year,400).eq.0)then
7              write(*,*)year,' is a leap year.'
8          else
9              write(*,*)year,' is not a leap year.'
10         endif
11     else
12         write(*,*)'The year ',year,' is a leap year.'
13     endif
14 else
15     write(*,*)'Entered year ',year,' is not a leap year.'
16 endif
17 pause
18 end
```

```
Enter a year:
2100
2100 is not a leap year.
PAUSE statement executed
To resume execution, type go. Other input will terminate the job.
```

E#24: WAP to find the highest of three numbers entered by the user using else...if...then statement & logical expression.

```
1      real a,b,c,highest
2      write(*,*) 'Enter three different real numbers:'
3      read(*,*) a,b,c
4      if(a .gt. b .and. a .gt. c) then
5          highest=a
6      else if(b .gt. a .and. b .gt. c) then
7          highest=b
8      else
9          highest=c
10     endif
11     write(*,*) 'Highest among three is ', highest
12     pause
13     end
```



do n I = start, stop, update

.....

.....

n continue

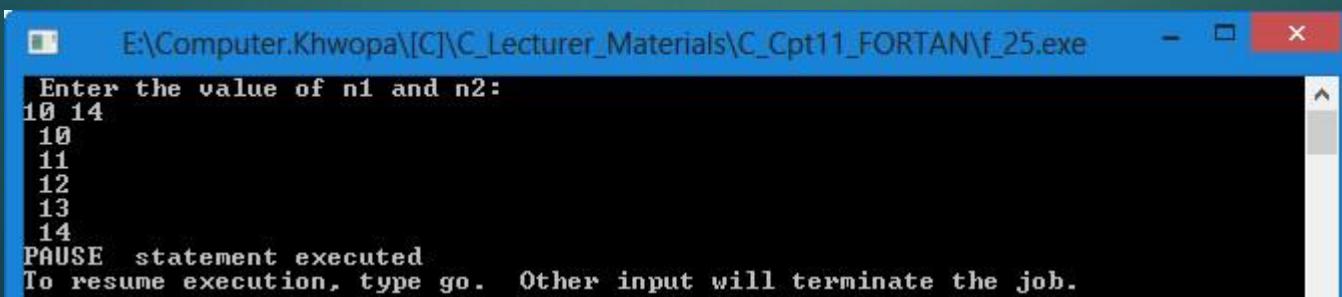
45

do loops

- E#25: Write a program to read two integers n1 and n2 ($n1 < n2$) and display the integers in the range inclusive.

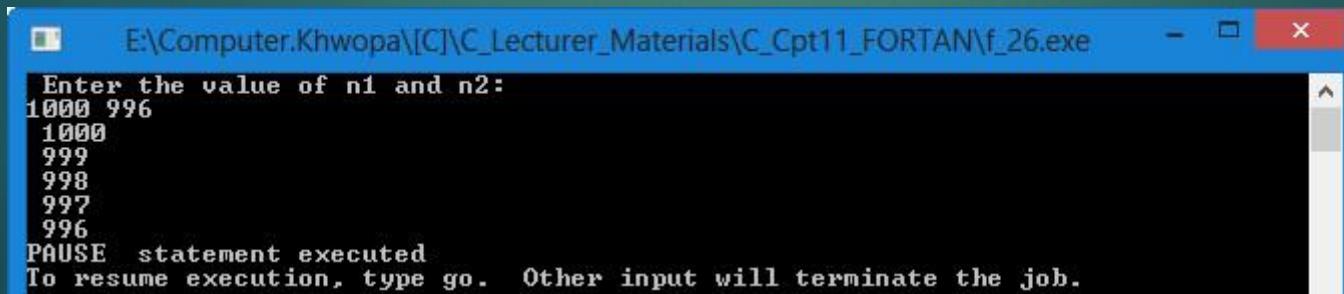
```
1      integer n1,n2,k
2      write(*,*) 'Enter n1 & n2:'
3      read(*,*) n1,n2
4      do 1 k=n1,n2,1
5      write(*,*) k
6      1      continue
7      pause
8      end
```

Er. Shiva K. Shrestha (HoD, Computer Department)
2019-02-17



E#26: Write a program to read two integers n2 and n1 ($n2 > n1$) and display the integers in the range from n2 to n1 inclusive.

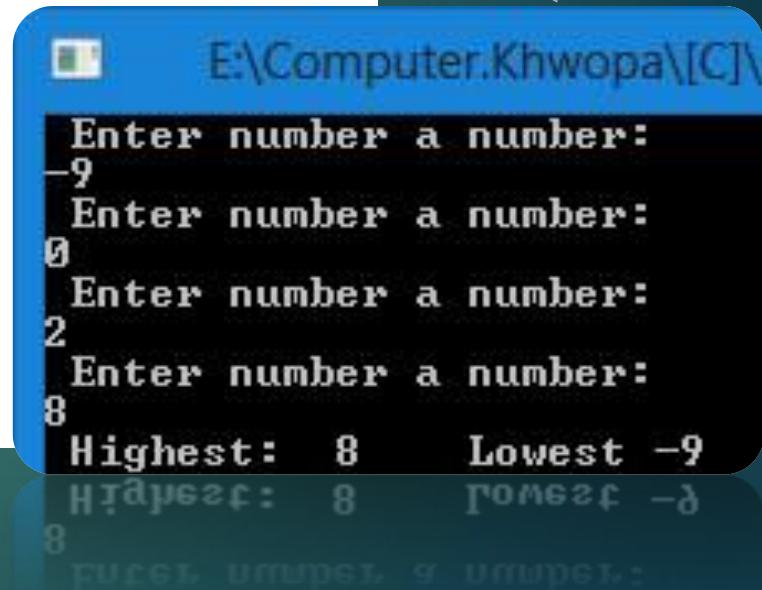
```
1      integer n2,n1,k
2      write(*,*) 'Enter n1 & n2:'
3      read(*,*) n2,n1
4      do 1 k=n2,n1,-1
5      write(*,*) k
6 1      continue
7      pause
8      end
```



```
E:\Computer.Khwopa\[C]\C_Lecturer_Materials\C_Cpt11_FORTAN\f_26.exe
Enter the value of n1 and n2:
1000 996
1000
999
998
997
996
PAUSE statement executed
To resume execution, type go. Other input will terminate the job.
```

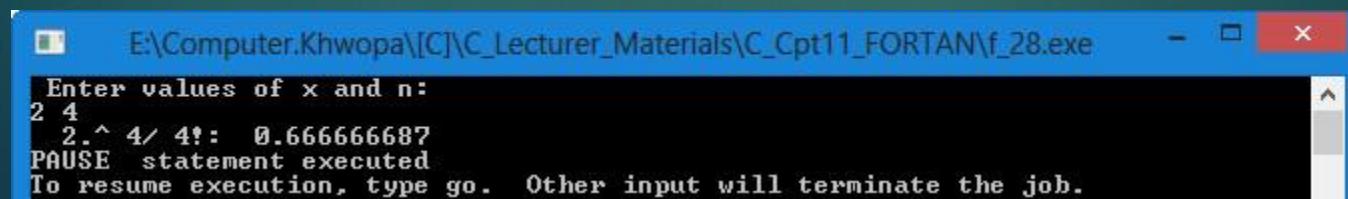
E#27: WAP to read four integers greater than zero & displays the highest and lowest among them.

```
1      integer n,h,l
2      h=0
3      l=0
4      do 1 k=1,4,1 !why k is not declared for its type here
5          write(*,*) 'Enter number a number:'
6          read(*,*) n
7          if(n.gt.h) then
8              h=n
9          endif
10         if(n.lt.l) then
11             l=n
12         endif
13     1 continue
14     write(*,*) 'Highest: ',h,' \tLowest',l
15     pause
16 end
```



E#28: WAP to read x as a real number and n as an integer and display the value of x^n & n!. Calculate x^n and n! using do loops.

```
1      integer n, f
2      real x, p
3      f=1
4      p=1
5      write(*,*) 'Enter x & n:'
6      read(*,*) x, n
7      do 1 k=1, n, 1
8          p=p*x
9      1      continue
10     do 2 k=1, n, 1
11         f=f*k
12     2      continue
13     write(*,*) x, '^', n, '/', n, '!:', p/f
14     pause
15     end
```



E#29 Evaluate:

$e^x = 1 + x + x^2 / 2! + x^3 / 3! + x^4 / 4! + x^5 / 5! \dots$ up to n terms.

```
1      write(*,*) 'Enter x & number of terms:'
2      read(*,*) x,n
3      term=1
4      do 1 k=1,n,1 !why x,k,n,term,sum are not declared?
5          term=term*x/k
6          sum=sum+term
7          write(*,*) 'term',k,':',term
8 1      continue
9      write(*,*) 'Sum of terms is:',sum
10     pause
11     end
```

```
Enter values of x and number of terms:
5 5
term 1: 5.
term 2: 12.5
term 3: 20.833334
term 4: 26.0416679
term 5: 26.0416679
Sum of terms is: 90.4166718
```

do n I = start, stop, update

.....

.....

do n1 I1 = start1, stop1, update1

.....

.....

continue

continue

```
1      integer sum,term,n
2      write(*,*) 'Enter number terms:'
3      read(*,*) n
4      sum=0
5      do 1 k=1,n,1
6          term=0
7          do 2 j=1,k,1
8              term=term+j
9              continue
10             sum=sum+term
11             write(*,*) 'term',k,term
12             continue
13             write(*,*) 'Sum of all terms:',sum
14             pause
15             end
```

estha (HoD, Computer Department)

```
E:\Computer.Khwopa>
Enter number terms:
4
term 1 1
term 2 3
term 3 6
term 4 10
Sum of all terms: 20
```

E#31: WAP to ask the user to enter two positive numbers f and s ($f > 1, s > 1$ and $f < s$). Display the prime numbers between the numbers inclusive.

```
1      integer f,s,k,check
2      write(*,*) 'Enter number first & second number:'
3      read(*,*) f,s
4      do 1 k=f,s,1
5          check=1
6          do 2 j=2,k/2,1
7              if(mod(k,j).eq.0)then
8                  check=0
9              endif
10         2 continue
11         if(check.eq.1)then
12             write(*,*) k, ' is prime.'
13         endif
14         1 continue
15         pause
16     end
```

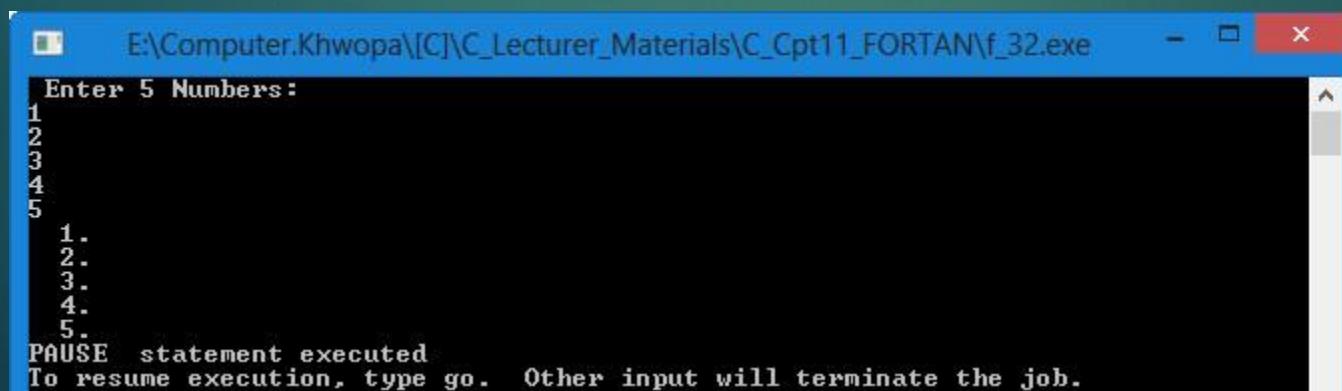
```
E:\Computer.Khwopa\[C]\C_Lecturer_Materials\C_Cpt11_FORTAN\f_31.exe
Enter number first and second number:
20 40
23 is prime.
29 is prime.
31 is prime.
37 is prime.
PAUSE statement executed
To resume execution, type go. Other input will terminate the job.
```

Array

1D Array

E#32 Reading & printing members of 1D array.

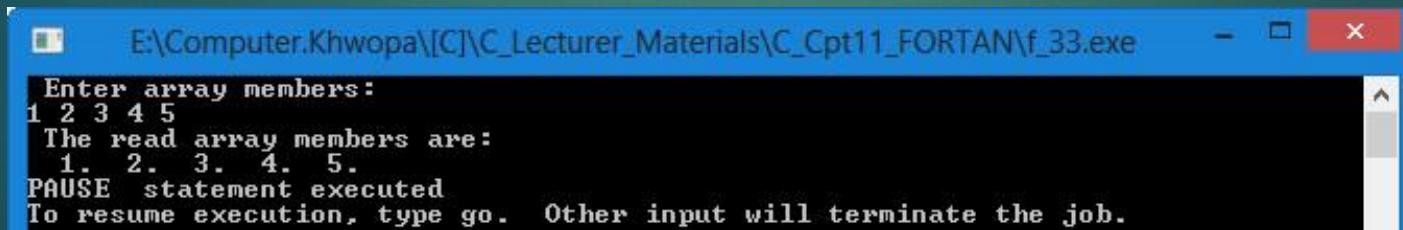
```
1      dimension a(5)
2      write(*,*) 'Enter 5 Numbers:'
3      do 1 k=1, 5, 1
4          read(*,*) a(k)
5 1    continue
6      do 2 k=1, 5, 1
7          write(*,*) a(k)
8 2    continue
9      pause
10     end
```



Implied Do

E#33: WAP to declare an array of size 5. Read & display array members using implied do loops.

```
1      dimension a(5)
2      write(*,*) 'Enter array members:'
3      read(*,*) (a(i),i=1,5)
4      write(*,*) 'The read array members are:'
5      write(*,*) (a(i),i=1,5)
6      pause
7      end
```

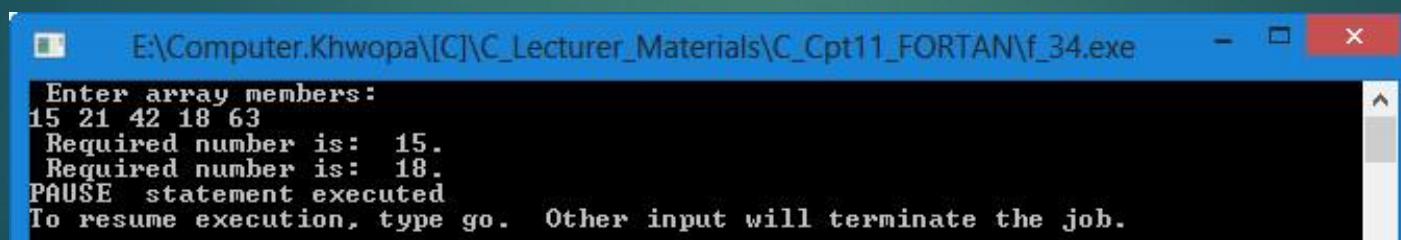


Normal do vs. implied do

Comparison between normal and implied do loop structure.			
	Normal do loop structure	Implied do loop structure	
1	do 1 k=1, 5, 1 read(*,*) a(k) continue	read(*,*) (a(i),i=1,5)	To read data for array members a(1), a(2), a(3), a(4), and a(5)
2	do 2 k=1, 5, 1 write(*,*) a(k) continue	write(*,*) (a(i),i=1,5)	To display content of array members a(1), a(2), a(3), a(4), and a(5)

E#34: WAP to read members of array of size 5 of type real and display those members which are exactly divisible by 3 but not by 7.

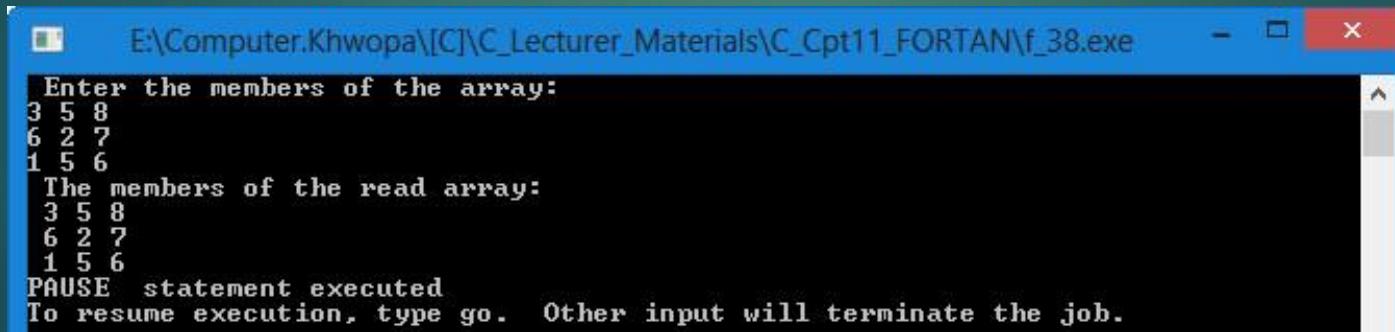
```
1      real a(5)
2      write(*,*) 'Enter array members:'
3      read(*,*) (a(i),i=1,5)
4      do 1 i=1, 5, 1
5          if(mod(a(i),3.0).eq.0.and.mod(a(i),7.0).ne.0)then
6              write(*,*) 'Required number is:',a(i)
7              endif
8 1      continue
9      pause
10     end
```



Array

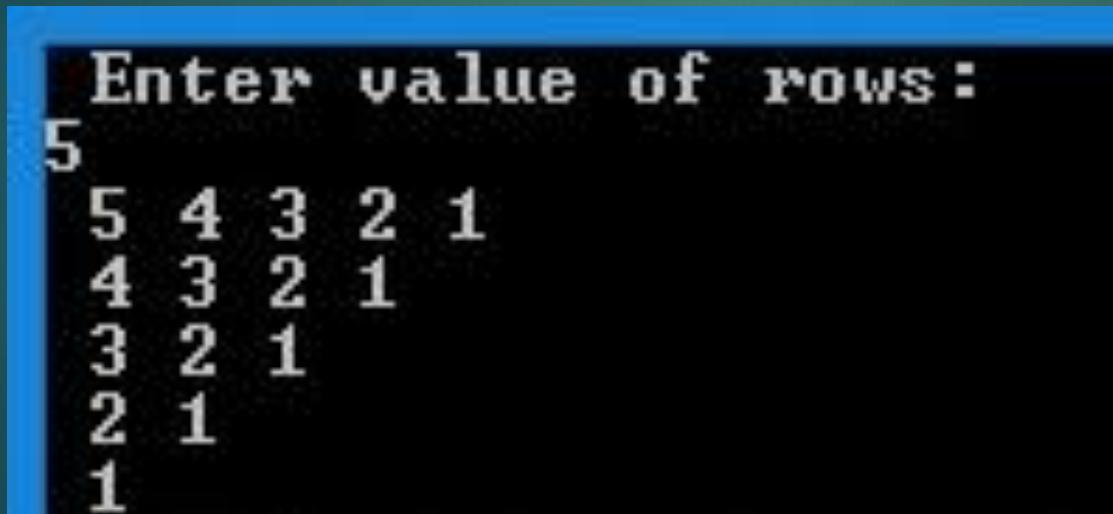
2D Array

```
1      integer x(3,3) ! two dimensional array declaration
2      write(*,*)'Enter 3*3 Matrix:'
3      do 1 i=1,3,1
4          read(*,*) (x(i,j),j=1,3,1) ! Reading 2D array row wise
5      1 continue
6      write(*,*)'The members of the read array:'
7      do 2 i=1,3,1
8          write(*,*) (x(i,j),j=1,3,1) ! Displaying 2D array row wise
9      2 continue
10     pause
11     End
```



42. WAP to print the following pattern of numbers. Ask the user to enter number of rows of the pattern.

```
1     write(*,*) 'Enter value of rows: '
2     read(*,*) i
3     do 1 k=i,1,-1
4         write(*,*)(j,j=k,1,-1) !implied do loop to print value of j
5 1    continue
6    pause
7    end
```



The image shows a terminal window with a blue header bar. The text inside the window is as follows:

```
Enter value of rows:
5
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

Q/A ?

Thank You!

Er. Shiva K. Shrestha
computer.khwopa@gmail.com