File pointer - analog screen cursor current position in the open file. When you open a file file pointer is always in the first position of the first term, then it moves depending on the requirements of operators or READ WRITE. Each new operator READ or WRITE begins next record (line). Office file pointer is also possible with the use aspect IO.

BACKSPASE operator moves the file pointer one line up. To set the file pointer in the first position of the first line used by the operator REWIND.

```
REWIND (UNIT = N), REWIND N
BACKSPASE (UNIT = N), BACKSPASE N
```

UNIT parameter specifies the device number by which he is identified in the program (a type INTEGER). The official word UNIT is optional.

Example. Save the file with the number 18 device variables A, B, C, and read the values of variables X, Y, Z.

```
WRITE (18, *) A, B, C
REWIND 18
READ (18, *) X, Y, Z
```

INQUIRE operator allows the app to determine the properties of the device (file). This statement has many opportunities and options. As an example, we show only the definition file name to the number of the device.

Example. Determine whether the device is open 16? If so, identify the name of the file and give it to the variable FNAME type CHARACTER.

```
INQUIRE (UNIT = 16, OPENED = OPN)
IF (OPN) THEN! Logical variable OPN = .TRUE.
INQUIRE (UNIT = 16, FILE = FNAME)
WRITE (*, *) '********* Opened File', FNAME
ELSE! Logical variable OPN = .FALSE.
WRITE (*, *) '********* Unit 16 do not open'
ENDIF
```

The operator makes an entry ENDFILE "EOF" in the current location of the file pointer.

```
ENDFILE N
```

N - number of the device.

After installing recording "EOF" ENDFILE operator sets the file pointer after writing the end of the file. This provides further serial data after the statement BACKSPACE or REWIND. ENDFILE can hide all records again for entries "end of file".

Logical function EOF (N) returns .TRUE., If the file pointer is on record "EOF" device number N and .FALSE., When no.

Example. Determine how many records in the file INPUT.TXT.

```
OPEN (1, FILE = 'INPUT.TXT', STATUS = 'OLD')
I = 0
DO WHILE (.NOT.EOF (1))
READ (1 *)
```

```
I = I + 1
END DO
WRITE (*, *) '*** Number of the records in INPUT.TXT:', I
```

Data exchange between software units is possible via the file interface. However, remember that the accuracy of data mapping file (on screen) is limited compared to the accuracy of the virtual safety during processing. While it dependent on hardware and software PC, but practically, you should count on seven significant digits for single-precision data.

Example. The above calculation program displacement U (m) drive that spins to realize data transfer function FN via file interface.

```
PROGRAM WH2
INTEGER H
REAL MU
DATA R1 / .05 /, R2 / .1`3 /, E /2.1E11/, G / 9.8 /, GAM / 7.7E4 /, H / 2 /, MU / .3 /
DATA W1, W2, DW / 50., 150., 10. /
WRITE (*, *) '********** INPUT DATA **********'
WRITE (*, *) 'R1 =', R1, 'R2 =', R2
WRITE (*, *) 'E =', E, 'G =', G
WRITE (*, *) 'GAM =', GAM, 'H =', H, 'MU =', MU
WRITE (*, *) '********** SOLVED NEXT **********'
WRITE (*, *) 'W (rad / c) U (m)'
OPEN (1, FILE = 1.TXT, STATUS = 'REPLACE')
WRITE (1, *) MU, GAM, H, G, E
OPEN (2, FILE = 2.TXT, STATUS = 'REPLACE')
DO W = W1, W2, DW
2 REWIND
WRITE (2, *) W
U = FN () * R2 / 8 * ((3 + MU) / (1-MU) * (R1 ** H + R2 ** H) +
      + (3 + MU) / (1-MU) * (R1 ** H-R2 ** H))
WRITE (*, *) W, U
ENDDO
CLOSE (1)
CLOSE (2)
STOP
CONTAINS
FUNCTION FN ()
REWIND 1
2 REWIND
READ (1, *) X1, X2, X4, X5, X6
READ (2, *) X3
FN = (1-X1 ** X4) * X2 * X3 ** X4 / X6 / X6
END FUNCTION FN
END
```