

File handling

Ref Pseudocode

Writing to a file

A file keeps data after the program ends. 0478 uses four commands, and every one names the file it works on.

```
OPENFILE <file identifier> FOR WRITE
WRITEFILE <file identifier>, <data>
CLOSEFILE <file identifier>
```

```
OPENFILE "names.txt" FOR WRITE
WRITEFILE "names.txt", "Mei"
WRITEFILE "names.txt", "Sam"
CLOSEFILE "names.txt"
OUTPUT "Saved"
```

Three **file modes** :

Mode	What it does
WRITE	start a new file — anything already there is lost
APPEND	add to the end of what is there
READ	read from the start

```
OPENFILE "log.txt" FOR WRITE
WRITEFILE "log.txt", "first"
CLOSEFILE "log.txt"
OPENFILE "log.txt" FOR APPEND
WRITEFILE "log.txt", "second"
CLOSEFILE "log.txt"
OUTPUT "Both lines saved"
```

- One WRITEFILE writes **one line**.
- CLOSEFILE is not optional. An unclosed file can lose the data still waiting to be written, and that is a stock exam answer.

Common mistakes

- FOR WRITE when you meant FOR APPEND, which silently destroys the old contents.
- Forgetting CLOSEFILE.
- Opening a file that is already open.

Reading a file to the end

READFILE takes one line into a variable. The variable must be declared first.

```
OPENFILE <file identifier> FOR READ
READFILE <file identifier>, <variable>
CLOSEFILE <file identifier>
```

Normally you do not know how many lines there are, so you read **until the end of the file** — which is what the EOF function reports.

```
OPENFILE "names.txt" FOR WRITE
WRITEFILE "names.txt", "Mei"
WRITEFILE "names.txt", "Sam"
WRITEFILE "names.txt", "Ana"
CLOSEFILE "names.txt"
```

```
DECLARE Line : STRING
OPENFILE "names.txt" FOR READ
WHILE NOT EOF("names.txt") DO
    READFILE "names.txt", Line
    OUTPUT Line
ENDWHILE
CLOSEFILE "names.txt"
```

EOF("names.txt") is TRUE once every line has been read, so WHILE NOT EOF(...) means *while there is still something to read*.

Counting while reading

```
OPENFILE "marks.txt" FOR WRITE
WRITEFILE "marks.txt", 40
WRITEFILE "marks.txt", 65
WRITEFILE "marks.txt", 88
CLOSEFILE "marks.txt"

DECLARE Mark, Count, Total : INTEGER
Count ← 0
Total ← 0
OPENFILE "marks.txt" FOR READ
WHILE NOT EOF("marks.txt") DO
    READFILE "marks.txt", Mark
    Count ← Count + 1
    Total ← Total + Mark
ENDWHILE
CLOSEFILE "marks.txt"
OUTPUT Count, " marks, mean ", Total / Count
```

Common mistakes

- A WHILE EOF(...) loop — that runs only when there is nothing left. You want NOT.
- Reading past the end because the loop has no EOF guard.
- Opening FOR READ a file that was never written.