



```

PROCEDURE FILLBUFFER;
BEGIN
    EMPTY := BLOCKREAD (INFILE,BUFFER,2) = 0;
    IF NOT EMPTY THEN BEGIN
        NOTNULLS := BUFSIZE + SCAN (- BUFSIZE, <> CHR(0),
            BUFFER [1023]);
        BUFINDEX := 0;
    END;
END;

PROCEDURE OPENFILE (FNAME: STRING);
BEGIN
    IF ((POS('.text',FNAME) = 0) AND
        (POS('.TEXT',FNAME) = 0)) THEN
        FNAME := CONCAT (FNAME, '.TEXT');
    RESET (INFILE, FNAME);
    FILLBUFFER; FILLBUFFER;
END;

PROCEDURE READFILE (VAR LINE: STRING);
VAR INDENT,
    LINELEN: INTEGER;
BEGIN
    IF BUFINDEX >= NOTNULLS THEN FILLBUFFER;
    IF NOT EMPTY THEN BEGIN
        LINELEN := SCAN (BUFSIZE, = CHR(13), BUFFER [BUFINDEX]);
        IF BUFFER [BUFINDEX] = CHR (16) THEN BEGIN
            INDENT := ORD (BUFFER [BUFINDEX + 1]) - 32;
            (*$R-*)
            LINE [0] := CHR (LINELEN + INDENT - 2);
            (*$R+*)
            IF INDENT > 0 THEN FILLCHAR (LINE [1], INDENT, ' ');
            IF LINELEN > 2 THEN MOVELEFT (BUFFER [BUFINDEX + 2],
                LINE [1 + INDENT], LINELEN - 2);
        END ELSE BEGIN
            (*$R-*)
            LINE [0] := CHR (LINELEN);
            (*$R+*)

            IF LINELEN > 0 THEN MOVELEFT (BUFFER [BUFINDEX],
                LINE [1], LINELEN);
        END;
    END;
END;

```

```

        BUFINDEX := BUFINDEX + LINELEN + 1;
    END;
END;

PROCEDURE GETOPTION;
BEGIN
    REPEAT
        WRITELN;
        WRITE ('OPTIONS: Q)uickread, R)eadln, E)xit ');
        CH := ' ';
        READ (KEYBOARD, CH);
    UNTIL CH IN ['Q', 'q', 'R', 'r', 'E', 'e'];
    IF CH IN ['E', 'e'] THEN BEGIN
        PAGE (OUTPUT);
        RESET (INFILE, FILENAME);
        CLOSE (INFILE, PURGE);
        EXIT (PROGRAM);
    END;
END;

PROCEDURE PRINTFILE;
BEGIN
    PAGE (OUTPUT); WRITELN;
    CASE CH OF
        'Q', 'q': BEGIN
            OPENFILE (FILENAME);
            REPEAT
                LINE := '';
                READFILE (LINE);
                WRITELN (LINE);
            UNTIL EMPTY;
            CLOSE (INFILE);
        END;
        'R', 'r': BEGIN
            RESET (TEXTFILE, FILENAME);
            REPEAT
                READLN (TEXTFILE, LINE);
                WRITELN (LINE);
            UNTIL EOF (TEXTFILE);
            CLOSE (TEXTFILE);
        END;
    END; (* CASE *)
END;

PROCEDURE INIT;
VAR TEMP: STRING;
BEGIN
    PAGE (OUTPUT); WRITELN;
    WRITE ('One momemt, please...');
    REWRITE (TEXTFILE, FILENAME);
    WRITE (TEXTFILE, 'This is a test file written to ');
    WRITE (TEXTFILE, 'demonstrate the ');
    WRITE ('.');

```

```

WRITELN (TEXTFILE, '"QUICKREAD" program. ');
WRITE ('. ');
WRITELN (TEXTFILE);
TEMP := 'I'm melting.';
REPEAT
    WRITE (TEXTFILE, 'TEST STRING --> ');
    WRITELN (TEXTFILE, TEMP);
    DELETE (TEMP, LENGTH (TEMP), 1);
    WRITE ('. ');
UNTIL TEMP = '';
WRITELN (TEXTFILE, ' AHHHHHHhhhhh.... ');
WRITELN (TEXTFILE);
WRITE (TEXTFILE, 'Notice that using READLN takes much ');
WRITE (TEXTFILE, 'more time');
WRITELN (TEXTFILE, ' to read a long line');
WRITE ('. ');
WRITELN (TEXTFILE, 'than it takes to read a short one. ');
WRITE ('. ');
CLOSE (TEXTFILE, LOCK);
WRITELN;
END;

```

```

BEGIN
    INIT; (* Write a temporary file for *)
    HELL_FREEZES_OVER := FALSE;
    REPEAT
        GETOPTIONS; (* Get users choice *)
        PRINTFILE; (* Print the file *)
    UNTIL HELL_FREEZES_OVER; (* Forever. *)
END.

```

Apple Tech Notes

Keywords: <None>

=====

This information is from the Apple Technical Information Library.

19960215 11:05:19.00

Tech Info Library Article Number: 653