DESIGN & DEVELOPMENT OF A NEW AND EFFICIENT APPROACH FOR LINE TEXT EDITING IN TURBO PASCAL 7 FOR WINDOWS*

K. J. Satao**

ABSTRACT:

Text editors come in the forms viz. Line editors, Stream editors, Screen editors, Word processors, Structure editors, etc. There are many text editors provided with Windows viz. Notepad, WordPad, Microsoft Office Word, etc. The MS-DOS editors viz. EDLIN, EDIT also work in Windows. But none of the above editors is interactive. The process of entering the programs/text shall be greatly enhanced, for the newcomers, if the editor is an interactive one. This paper gives the design & development of a new and efficient approach for line text editing, the most basic and fundamental editing, in Turbo Pascal 7 for Windows OS. It is proposed to give a new, interactive, and more user friendly approach to line text editing for document / non document(program) files using singly linked lists. A menu driven program is designed and developed. The program displays all the options and once a user chooses a particular option, he is prompted further interactively. The program is fully tested on Windows XP, with total RAM = 256 MB. This research work has, in all, twenty three options and is academically (not available in any book) very useful for the Computer Science & Engineering and Information Technology disciplines.

Keywords: Authoring tools and methods; Computer-mediated communication; Human-computer interface; Interactive learning environments; Programming and programming languages.

** Professor & Head, Computer Science and Engineering, Rungta College of Engg. & Technology, Kohka Road, Kurud, Bhilai-490 024, Chhattisgarh, INDIA.

1. INTRODUCTION

The options in the research work are…"C" for clearing the screen, "A" for appending lines, "L" for listing from a line number to a line number, "E" for listing the entire file, "I" for inserting lines before a line number, "D" for deleting from a line number to a line number, "U" for updating a line, "W" for writing entire text in the file opened or currently created by the user, "F" for writing specific lines in a specific file, "O" for copying from a line number to a line number before a line number, "M" for moving from a line number to a line number before a line number, "S" for searching a string, "R" for replacing a string, "G" for listing page wise(desired number of lines at a time), "V" for converting upper case to lower case and vice-versa or to title/sentence case, from a line number to a line number, "T" for copying from another file before a line number, "P" for printing a file with or without heading, "Y" for displaying directory contents, "K" for deleting files, "J" for searching archive files in a directory for a string, "H" for displaying or modifying file attributes, "N" for opening a new file, and "Q" for quitting.

This research work has the following main features...

1. No user training is required. A user can use FDD / HDD / CR-ROM drive / Pen drive.
2. No referring of the manual for the options is required. The program gives starting and ending times.
3. It is very useful in the students' environment i.e. in the schools and colleges where a batch of students may come for the practical work and a single instructor/teacher is supposed to look after the batch.
4. This work is compatible with text processing utilities such as EDLIN, EDIT, Notepad, etc.
5. A file can be created in or saved in or loaded from any directory. Program displays file attributes.
6. Insertion/appending/updation is made very easy. The program takes care of RO, H, S, D, A, Vol Id. files.
7. Each option gives a starting message so that the user becomes sure about what he shall be doing and an ending message so that he shall become sure about the completion of the desired task.
8. A user shall be able to update files even with the .bak extension. A backup file with an extension of .OLD is created before every overwriting of the file.

* Orally presented and demonstrated in the National Conference on "Advanced Computing Techniques", "BITCON – 2007", organized by Bhilai Institute of Technology, Durg, C.G., India, held during 16th & 17th March 2007.

2. IMPORTANT PROCEDURES AND MESSAGES INCORPORATED

[Modified, Windows versions of Satao, 1992 and 1994]

2.1 Procedure USAGE

```
'An Interactive Line Text Editor for Windows using Pascal 7.0 By Prof. K.J.Satao'
'Usage : WPEDITOR<Enter> from the prompt or'
'        Double Click on WPEDITOR Icon or'
'        Select WPEDITOR Icon and Hit Enter key'
'(Alt & Enter keys give full screen view, pressing again gives title bar)'
```

2.2 Procedure GET_DRIVE_NAME

```
'Please enter Floppy/Hard Disk/CD-ROM/Pen drive name...A/B/C/D/E/F/G/H etc. '
```

2.3 Procedure GET_FILE_NAME_MESSAGE

```
'Please give file name...'
'You may enter just file name(8+3) for current directory'
'Or you may  enter \file name(8+3) for root directory file'
'Or give path & file name e.g. PASCAL7\WPEDITOR.PAS'
'78 characters maximum(case insensitive e.g. TeST.c = TEST.C) '
```

2.4 Procedure DIS_MEM_CAP; { DISPLAY MEMORY CAPACITY }

```
'Main memory available = ', MEMAVAIL, ' Bytes'
```

2.5 Procedure GET_FILE_NAME2

```
GET_FILE_NAME_MESSAGE
'You may hit just Enter key for *.* in current directory'
'You can use *, ? as wild characters(? for single character). '
```

2.6 Welcome Message

```
'File:FILE_NAME                              Date:DD/MM/YYYY Day:DAY Time:HH:MM:SS'
'-------------------------------------------------------------------------------'
'Welcome to An Interactive Line Text Editor by Prof. K. J. Satao,Bhilai(CG),India'
'-------------------------------------------------------------------------------'
'Please choose an option...C-Clear screen/A-Append/L-List specific lines/E-Entire'
'list/I-Insert/D-Delete/U-Update/W-Write in open file/F-Write in any file/O-Copy/'
'M-Move/G-Pagewise list/V-Convert case/T-Copy from other file/S-Search/R-Replace/'
'P-Print a file/Y-Directory/K-Delete files/H-Change attributes/J-Search in files/'
'N-Open new file/Q-Quit(Upper/Lower Case) '
```

2.7 Procedure NUMBER_CHECK

```
IF (NUMBER = 0) WRITE 'Error in number, Value = 0, New value = 1'
IF (NUMBER < 0) WRITE 'Error in number, Value is negative, New value = 1'
```

2.8 Procedure APPEND_INSERT_UPDATE_MESSAGE

```
'Use F1 or --> to take a character from the  previous  line in the current line.'
'Use F2 or <-- or Del to delete a character from previous line for current line.'
'(F2 or <-- or Del does not  physically  delete a character from previous line).'
'Use F3 or End to take  all  the  remaining  characters from the  previous line.'
'Tab key inserts 5 blank characters. Backspace key  deletes  previous character.'
```

2.9 Insert_Append Message

```
'You can have up to 132 characters per line. End each line with Enter key.'
'New line begins automatically after 132 characters are entered in a line.'
'Use Ctrl d/D to end appending/inserting. You can have maximum 3455 lines.'
APPEND_INSERT_UPDATE_MESSAGE
'Save(W/F) your work at regular intervals to avoid the accidental loss of data.'
```

2.10 Update Message

```
'You can have  maximum  132 characters in the line. End the line with Enter key.'
APPEND_INSERT_UPDATE_MESSAGE
'Updation  automatically  stops after 132 characters are  entered  in the line.'
```

2.11 Procedure GET_LINES

```
'From which line number ? '(Say NUMBER)
'To   which line number ? '(Say STOP)
```

2.12 Delete Message

```
'Please note...this option will delete the desired lines.'
'The deleted lines cannot be recovered later on unless the lines are written in a'
'file before proceeding.'
'Do you surely want to delete line(s) ? (Y/y for yes) '
```

2.13 Convert Case Message

```
'Please enter...L/l for lower case or U/u for upper case or T/t for title case or'
'S/s for sentence case or Hit Enter key to abort '
```

2.14 Procedure SEARCH_OPTIONS

```
'Do you want to ignore case ? (Y/y for yes) '
'? can be used as a wild character e.g. s???o shall match with satao, spqro, etc.'
'Be careful in using it...Use it ? (Y/y for yes) '
'Which string ? (First 20 characters are considered...Hit just Enter to abort)'
```

2.15 Replace Message

```
SEARCH_OPTIONS
'By which string ? (First 20 characters are considered)'
'Replace by query ? (Y/y for yes) '
```

2.16 Procedure PRINT_FILE

```
'Printing a file on parallel printer i.e. LPT1/PRN'
GET_FILE_NAME1
'Please save your work before proceeding...Runtime error may occur...'
'Choose...1..To print or 2..To abort '(Say I)
'Choose...1..For 80 column printer or 2..For 132 column printer '(Say J)
'Do you want page heading ? (Y/y for yes) '(Say CH1)
```

2.17 Procedure NEW_FILE

```
'Please note...option will delete all current lines.'
'The deleted lines cannot be recovered later on.'
'You should proceed only after writing the current lines in a file.'
'Do you surely want to proceed ? (Y/y for yes) '
```

3. PROPOSED FURTHER RESEARCH WORK

It is proposed to use doubly linked lists, circular linked lists and develop interactive line text editors and screen editors. It is proposed to remove the present limitations of the program viz. Windows uses up to 256 characters file name whereas the current program uses 8+3 format, Windows mostly uses USB printer whereas the program uses parallel printer.

4. ACKNOWLEDGEMENTS

The author is highly thankful to Hon'ble Shri Santosh Rungta, Respected Shri Sourabh Rungta, and Respected Shri Sonal Rungta of Rungta College of Engineering & Technology, Bhilai for providing the necessary facilities for the work.

5. REFERENCES

1. Satao K. J., 1992, An Interactive Line Text Editor for MS-DOS Operating System, Indian Computing Congress, 1992, Hyderabad, India, held during Dec. 17-19, 1992; published in the proceedings of the conference, ISBN : 0-07-462029-0, titled "Indian Computing Congress Series - Innovative Applications in Computing" edited by E. Balgurusamy & B. Sushila, published by Tata McG Hill, at P. P. 439-460.

2. Satao K. J., 1994, An Interactive Line Text Editor for Novell's Netware Operating System, National Conference - "Frontier'94" at Babasaheb Naik College of Engineering, Pusad(M.S.), India, held during Feb. 1-2, 1994; published in the proceedings of the conference at P. P. 131-145.

APPENDIX: THE SOURCE CODE OF THE PROGRAM

[Modified, Windows version of Satao, 1992 and 1994]

```
{* Design  and  Development of a  New and Efficient  Approach(Interactive) for *
 * Line Text Editing using Linked Lists in  Pascal Language for Windows, O.S., *
 * WPEDITOR.PAS,by Prof. K.J.Satao,Rungta College of Engineering & Technology, *
 * Kohka Road, Kurud,Bhilai-490 024,C.G.,India. Compiler used is Turbo Pascal, *
 * Ver 7. Tested on Microsoft  Windows  XP  Professional with  total  Physical *
 * Memory = 256 MB. Available  Physical Memory = 35.93 MB. Build  using  Large *
 * Memory sizes and run the executable program. For Memory sizes- Choose Alt O *
 * from the IDE : Options-Memory sizes-Stack size = 16384, High Heap = 655360. *}
PROGRAM WPEDITOR; { Final version }
USES CRT, STRINGS, WINDOS;
LABEL 1, 5, 10, 15, 20, 25;
CONST
 RESERVED_FILE_NAMES : ARRAY[1..13] OF PCHAR = ('AUX','CLOCK$','COM1','COM2',
  'COM3','COM4','CON','LPT1','LPT2','LPT3','LST','NUL','PRN');
 ALLOWED_FILE_NAME_CHARS : SET OF CHAR = ['0'..'9','A'..'Z','a'..'z','.','\','*',
  '?','(',')','_','^','$','~','!','#','%','&','-','{','}','@',CHR(ORD(39)),' '];
 WORD_SEPARATORS : SET OF CHAR = ['`','~','!','@','#','$','%','^','&','*','(',')',
  '-','_','+','=','{','[','}',']','|','\',':',';','"',',','<','.','>','/','?',
  ' '];
 SENTENCE_SEPARATORS : SET OF CHAR = ['!','@','(','{','[',':','<','.','?'];
 DAY_NAME : ARRAY[0..6] OF STRING[3]=('SUN','MON','TUE','WED','THU','FRI','SAT');
TYPE
 TEXT1    = ARRAY[1..134] OF CHAR; { MAXIMUM LINE LENGTH = 132 CHARS + CR + LF }
 STRING12 = ARRAY[1..20]  OF CHAR; { SEARCH/REPLACE 20 CHARACTERS AT A TIME }
 CLEAR    = ARRAY[1..50]  OF CHAR; { ARRAY OF LF i.e. CHR(10) FOR CLEAR SCREEN }
 POINTER1 = ^ENTRY;                { FIRST LINKED LIST FOR MAIN TEXT }
 ENTRY    = RECORD
             LINE_NO   : INTEGER;
             LINE_TEXT : TEXT1;
             NEXT      : POINTER1;
            END;
 POINTER2 = ^ENTRY1;               { SECOND LINKED LIST FOR COPY / MOVE LINES }
 ENTRY1   = RECORD
             LINE_NO1   : INTEGER;
             LINE_TEXT1 : TEXT1;
             NEXT1      : POINTER2;
            END;
 PATH = ARRAY[0..FSPATHNAME] OF CHAR; DIR = ARRAY[0..FSDIRECTORY] OF CHAR;
 NAME = ARRAY[0..FSFILENAME] OF CHAR; EXT = ARRAY[0..FSEXTENSION] OF CHAR;
VAR
 IGNORE_CASE, QUERY, WILD, OPTION, CHOICE, CH, OLD_CH, OVER_FLOW, SAVE : CHAR;
 FILE_NAME, OLD_FILE_NAME, ASCIIZ, O_ASCIIZ, F_NAME : PATH;
 NUMBER, START, STOP, HIGHEST_NO, HIGHEST_NO1, CODE, LINE_OCC, DR_NO : INTEGER;
 NEW_LINE, NEW_LINE1 : TEXT1; F, PRINTER : TEXT; DR_NAME : ARRAY[0..0] OF CHAR;
 HEAD, LINK, PREV, P : POINTER1; TEMP, TEMP1, TEMP2, B : TEXT1;  CLEAR1 : CLEAR;
 S1 : PATH; S2 : STRING; BUF : ARRAY[1..10240] OF CHAR; DIRINFO : TSEARCHREC;
 HEAD1, LINK1, PREV1, P1 : POINTER2; TOT_CH, TOT_OCC, REP_DONE : LONGINT;
 DONE, INVALID_FILE_NAME, AVAILABLE_MEMORY, SAVED, SEN_SEP : BOOLEAN;
 TEMP12, TEMP14, B12, B13, REP_STRING : STRING12; D1 : DIR; N1 : NAME; E1 : EXT;
 M, N, I, J, K, I1, J1, K1, NU, NU1, OCC, L, L1, FVAL, SNO, LINE_REP : INTEGER;
 DAY, MONTH, YEAR, DAYOFWEEK, HOURS, MINUTES, SECONDS, HUNDREDTHS : WORD;
 HOURS1, MINUTES1, SECONDS1, HOURS2, MINUTES2, SECONDS2 : WORD; A : STRING[5];
 ATTR : WORD; HOURS11, MINUTES11, SECONDS11 : INTEGER; DAY_NAM : STRING[3];
 PAGE_HEADING_CHARS, PAGE_HEADING_LINES, T_CH, LINE_OCC1, LINENUM, PGNO : INTEGER;
```

```
PROCEDURE SOUND1;
BEGIN
 SOUND(1000 + RANDOM(2000)); DELAY(200); NOSOUND;
END;
PROCEDURE SPLIT_FILE_NAME;
BEGIN
 I := 0;
 WHILE (FILE_NAME[I] <> #0) DO
 BEGIN { SMALL LETTERS ARE CONVERTED INTO CAPITAL LETTERS }
  CH := FILE_NAME[I];
  IF ((ORD(CH) >= 97) AND (ORD(CH) <= 122)) THEN
  FILE_NAME[I] := CHR(ORD(FILE_NAME[I]) - 32); I := I + 1;
 END;
 FILESPLIT(FILE_NAME, D1, N1, E1);
END; { D1 = DIRECTORY(67), N1 = PRIMARY NAME(8), E1 = EXTENSION(.+3) }
PROCEDURE TEST_FILE_NAME;
BEGIN
 INVALID_FILE_NAME := FALSE; TEXTCOLOR(CYAN);
 IF (N1[0] = #0{NULL}) THEN
 BEGIN
  WRITELN; WRITELN('Invalid file name...Primary file name cannot be null');
  INVALID_FILE_NAME := TRUE; TEXTCOLOR(YELLOW); EXIT;
 END;
 FOR I := 1 TO 13 DO
 IF ((STRCOMP(N1, RESERVED_FILE_NAMES[I]) = 0) AND (E1[1] = #0{''})) THEN
 BEGIN { CHECKING WHETHER THE PRIMARY FILE NAME IS A RESERVED WORD OR NOT }
  WRITELN; WRITELN('Invalid file name...It cannot be a reserved word');
  INVALID_FILE_NAME := TRUE; TEXTCOLOR(YELLOW); EXIT;
 END;
 I := 0;
 WHILE (N1[I] <> #0) DO
 BEGIN
  IF (NOT(N1[I] IN ALLOWED_FILE_NAME_CHARS)) THEN
  BEGIN
   WRITELN;
   WRITELN('Invalid file name...Invalid character... ', N1[I],
           ' ...in primary file name');
   INVALID_FILE_NAME := TRUE; TEXTCOLOR(YELLOW); EXIT;
  END;        { CHECKING WHETHER THE PRIMARY FILE NAME CONTAINS THE PERMITTED }
  I := I + 1; { CHARACTERS OR NOT }
 END;
 IF (E1 = ''{NULL}) THEN BEGIN TEXTCOLOR(YELLOW); EXIT; END;
 I := 1; { DOT(.) IS A PART OF EXTENSION, HENCE START WITH 1 }
 WHILE (E1[I] <> #0) DO
 BEGIN
  IF (NOT(E1[I] IN ALLOWED_FILE_NAME_CHARS)) THEN
  BEGIN
   WRITELN;
   WRITELN('Invalid file name...Invalid character... ', E1[I],
           ' ...in secondary file name');
   INVALID_FILE_NAME := TRUE; TEXTCOLOR(YELLOW); EXIT;
  END;        { CHECKING WHETHER THE SECONDARY FILE NAME CONTAINS THE PERMITTED }
  I := I + 1; { CHARACTERS OR NOT }
 END;
 TEXTCOLOR(YELLOW);
END;
```

```
PROCEDURE DISPLAY_FILE_ATTRIBUTES;
BEGIN
 ASSIGN(F, F_NAME); GETFATTR(F, ATTR); TEXTCOLOR(CYAN);
 IF (DOSERROR <> 0) THEN WRITE('Error code = ', DOSERROR)
 ELSE
 BEGIN
  IF (ATTR AND FAARCHIVE   <> 0) THEN WRITE('<Archive>');
  IF (ATTR AND FAREADONLY  <> 0) THEN WRITE('<Read Only>');
  IF (ATTR AND FAHIDDEN    <> 0) THEN WRITE('<Hidden>');
  IF (ATTR AND FASYSFILE   <> 0) THEN WRITE('<System>');
  IF (ATTR AND FADIRECTORY <> 0) THEN WRITE('<Directory>');
  IF (ATTR AND FAVOLUMEID  <> 0) THEN WRITE('<Volume Id>');
 END;
 WRITELN; TEXTCOLOR(YELLOW);
END;
PROCEDURE DIS_OPEN_FILE; { DISPLAY OPEN FILE }
BEGIN
 F_NAME := FILE_NAME; WRITELN; FINDFIRST(FILE_NAME, FAANYFILE, DIRINFO);
 TEXTCOLOR(CYAN); WRITELN('Open file name = ', FILE_NAME);
 WRITELN; TEXTCOLOR(YELLOW);
 IF (DOSERROR = 0) THEN
 BEGIN
  WRITE('Open file size = ', DIRINFO.SIZE, ' Bytes, Attributes...');
  DISPLAY_FILE_ATTRIBUTES;
 END;
END;
PROCEDURE GET_DRIVE_NAME;
BEGIN
 REPEAT
  WRITELN; WRITE('Please enter Floppy/Hard Disk/CD-ROM/Pen drive name...',
                 'A/B/C/D/E/F/G/H etc. ');
  SOUND1; CH := READKEY; DR_NAME[0] := UPCASE(CH);
  WRITE(DR_NAME);
 UNTIL (ORD(DR_NAME[0]) >= 65) AND (ORD(DR_NAME[0]) <= 90);
END;
PROCEDURE GET_PATH;
LABEL 10;
BEGIN
 STRCOPY(O_ASCIIZ, ASCIIZ); M := LENGTH(S2) - 1;
 IF ((L = 0) AND (S1[M] <> '\')) THEN
 BEGIN STRCOPY(ASCIIZ, S1); STRCAT(ASCIIZ, '\*.*'); GOTO 10; END;
 IF ((L = 0) AND (S1[M] = '\')) THEN
 BEGIN STRCOPY(ASCIIZ, S1); STRCAT(ASCIIZ, '*.*'); GOTO 10; END;
 IF ((D1[0] = #0) AND (S1[M] <> '\')) THEN
 BEGIN
  STRCOPY(ASCIIZ, S1); STRCAT(ASCIIZ, '\'); STRCAT(ASCIIZ, O_ASCIIZ); GOTO 10;
 END;
 IF ((D1[0] = #0) AND (S1[M] = '\')) THEN
 BEGIN
  STRCOPY(ASCIIZ, S1); STRCAT(ASCIIZ, O_ASCIIZ); GOTO 10;
 END;
 IF ((D1[0] = '\') AND (S1[M] <> '\')) THEN
 BEGIN
  FOR I := 0 TO 79 DO ASCIIZ[I] := #0;
  ASCIIZ[0] := DR_NAME[0]; STRCAT(ASCIIZ, ':'); STRCAT(ASCIIZ, O_ASCIIZ); GOTO 10;
 END;
```

```
 IF ((D1[0] = '\') AND (S1[M] = '\')) THEN
 BEGIN
  STRCOPY(ASCIIZ, S1); STRCAT(ASCIIZ, N1); STRCAT(ASCIIZ, E1); GOTO 10;
 END;
 IF (ASCIIZ[L-1] = '\') THEN
 BEGIN STRCAT(ASCIIZ, '*.*'); GOTO 10; END;
 FOR I := 0 TO 79 DO ASCIIZ[I] := #0;
 ASCIIZ[0] := DR_NAME[0]; STRCAT(ASCIIZ, ':\'); STRCAT(ASCIIZ, O_ASCIIZ);
10:STRCOPY(FILE_NAME, ASCIIZ); SPLIT_FILE_NAME; TEST_FILE_NAME;
 STRCOPY(ASCIIZ, FILE_NAME);
END;
PROCEDURE CHECK_ERROR;
BEGIN
 I := DOSERROR; J := 0; TEXTCOLOR(CYAN); WRITELN;
 IF (I = 3) THEN BEGIN WRITELN('Error...Path not found'); WRITELN; END
 ELSE IF (I = 6)  THEN BEGIN WRITELN('Error...Invalid handle'); WRITELN; END
 ELSE IF (I = 8)  THEN BEGIN WRITELN('Error...Not enough memory'); WRITELN; END
 ELSE IF (I = 10) THEN BEGIN WRITELN('Error...Invalid environment'); WRITELN; END
 ELSE IF (I = 11) THEN BEGIN WRITELN('Error...Invalid format'); WRITELN; END;
 TEXTCOLOR(YELLOW);
 IF ((I = 3) OR (I = 6) OR (I = 8) OR (I = 10) OR (I = 11)) THEN
 BEGIN
  WRITE('Retry...Hit any key to continue ');
  SOUND1; CH := READKEY; WRITELN; J := 1;
 END;
END;
PROCEDURE GET_FILE_NAME_MESSAGE;
BEGIN
 WRITELN; DR_NO := ORD(DR_NAME[0]) - 64;
 WRITELN('You may enter just file name(8+3) for current directory');
 WRITELN('Or you may  enter \file name(8+3) for root directory file');
 WRITELN('Or give path & file name e.g. PASCAL7\WPEDITOR.PAS');
 WRITE('78 characters maximum(Case insensitive e.g. TeSt.c = TEST.C) ');
END;
PROCEDURE GET_FILE_NAME1;
BEGIN
 GET_FILE_NAME_MESSAGE; SOUND1; READLN(ASCIIZ);
 {FILENAME CANNOT HAVE \ / : * " < > | ? + etc. }
 L := 0; WHILE (ASCIIZ[L] <> #0) DO L := L + 1;
 IF (L = 0) THEN
 BEGIN
  WRITELN; TEXTCOLOR(CYAN);
  WRITELN('Invalid file name...File name cannot be null');
  INVALID_FILE_NAME := TRUE; TEXTCOLOR(YELLOW); EXIT;
 END;
 FOR I := 0 TO L DO
 IF ((ASCIIZ[I] = '*') OR (ASCIIZ[I] = '?')) THEN
 BEGIN
  INVALID_FILE_NAME := TRUE; WRITELN; TEXTCOLOR(CYAN);
  WRITELN('Invalid file name...File name cannot have * or ?');
  TEXTCOLOR(YELLOW); EXIT;
 END;
 FILESPLIT(ASCIIZ, D1, N1, E1); GETDIR(DR_NO, S2);
 FOR I := 0 TO LENGTH(S2) DO S1[I] := S2[I];
 FOR I := 1 TO LENGTH(S2) + 1 DO S1[I-1] := S1[I]; GET_PATH;
END;
```

```
PROCEDURE GET_CHECK_FILE_NAME1;
LABEL 5;
BEGIN
5:GET_DRIVE_NAME; WRITELN;
 REPEAT GET_FILE_NAME1; UNTIL NOT(INVALID_FILE_NAME);
 FINDFIRST(FILE_NAME, FAANYFILE, DIRINFO);
 CHECK_ERROR; IF (J = 1) THEN GOTO 5;
END;
PROCEDURE GET_FILE_NAME2;
BEGIN
 GET_FILE_NAME_MESSAGE; WRITELN; WRITELN;
 WRITELN('You may hit just Enter key for *.* in current directory');
 WRITE('You can use *, ?(? for single character) as wild characters ');
 SOUND1; { FILENAME CANNOT HAVE \ / : " < > | + etc. }
 READLN(ASCIIZ); L := 0;
 WHILE (ASCIIZ[L] <> #0) DO L := L + 1;
 FILESPLIT(ASCIIZ, D1, N1, E1); GETDIR(DR_NO, S2);
 FOR I := 0 TO LENGTH(S2) DO S1[I] := S2[I];
 FOR I := 1 TO LENGTH(S2) + 1 DO S1[I-1] := S1[I];
 GET_PATH;
END;
PROCEDURE GET_CHECK_FILE_NAME2;
LABEL 5;
BEGIN
5:GET_DRIVE_NAME; WRITELN;
 REPEAT GET_FILE_NAME2; UNTIL NOT(INVALID_FILE_NAME);
 FINDFIRST(FILE_NAME, FAANYFILE, DIRINFO);
 CHECK_ERROR; IF (J = 1) THEN GOTO 5;
END;
PROCEDURE GET_NO_OF_CHARS; { GET NUMBER OF CHARACTERS }
BEGIN
 I := 135; REPEAT I := I - 1; UNTIL (TEMP[I] <> CHR(0));
END;
PROCEDURE DIRECTORY_HEADING;
BEGIN
 TEXTCOLOR(CYAN); IF (OPTION = 'Y') THEN WRITELN(CLEAR1) ELSE WRITELN;
 WRITELN('Directory : ', ASCIIZ); TEXTCOLOR(YELLOW);
 WRITE('-------------------------------------------------------------------------',
       '---------');
 WRITELN('S.N.    File Name       Size    Attributes...');
 WRITE('-------------------------------------------------------------------------',
       '---------');
END;
PROCEDURE DISPLAY_DIRECTORY;
VAR
 F : FILE; COUNT : INTEGER;
BEGIN
 IF (OPTION = 'Y') THEN GET_CHECK_FILE_NAME2;
 COUNT := 0; DIRECTORY_HEADING; FINDFIRST(ASCIIZ, FAANYFILE, DIRINFO);
 { ALL TYPES OF FILES...RO, H, S, D, A, AND VOL ID ARE LISTED }
 WHILE (DOSERROR = 0) DO
 BEGIN
  TEXTCOLOR(CYAN); COUNT := COUNT + 1;
  WRITE(COUNT:4, ' ', DIRINFO.NAME:12, '  ', DIRINFO.SIZE:10, '    ');
  STRECOPY(STRECOPY(F_NAME, D1), DIRINFO.NAME); DISPLAY_FILE_ATTRIBUTES;
  I := COUNT DIV 20;
```

```
  IF (COUNT = I * 20) THEN
  BEGIN
   WRITE('-----------------------------------------------------------------',
         'Hit any key...');
   SOUND1; CH := READKEY; TEXTCOLOR(YELLOW); DIRECTORY_HEADING;
  END;
  FINDNEXT(DIRINFO);
 END;
 WRITE('-----------------------------------------------------------------',
       'Hit any key...');
 SOUND1; CH := READKEY; TEXTCOLOR(YELLOW); WRITELN;
 WRITELN; WRITELN('Total number of files = ', COUNT);
END;
PROCEDURE DIS_FILE_NAMES; { DISPLAY FILE NAMES }
BEGIN
 FILESPLIT(FILE_NAME,D1,N1,E1);
 STRECOPY(STRECOPY(STRECOPY(ASCIIZ, D1), N1), '*.*');
 DISPLAY_DIRECTORY; DIS_OPEN_FILE;
END;
PROCEDURE DIS_MEM_CAP; { DISPLAY MEMORY CAPACITY }
BEGIN
 TEXTCOLOR(CYAN); WRITELN;
 WRITELN('Main memory available = ', MEMAVAIL, ' Bytes'); TEXTCOLOR(YELLOW);
END;
PROCEDURE LIST_LINES(START, STOP : INTEGER);
BEGIN
 TOT_CH := 0; DONE := FALSE; LINK := HEAD^.NEXT;
 WHILE (LINK <> NIL) AND NOT DONE DO
 IF (LINK^.LINE_NO > STOP) THEN DONE := TRUE
 ELSE IF (LINK^.LINE_NO >= START) THEN
 BEGIN
  IF ((OPTION <> 'W') AND (OPTION <> 'F')) THEN WRITE(LINK^.LINE_NO:4, ':');
  TEMP := LINK^.LINE_TEXT; GET_NO_OF_CHARS;
  FOR I1 := 1 TO I DO
  BEGIN
   CH := TEMP[I1]; IF ((OPTION <> 'W') AND (OPTION <> 'F')) THEN WRITE(CH);
   TOT_CH := TOT_CH + 1;
  END;
  LINK := LINK^.NEXT;
 END ELSE LINK := LINK^.NEXT
END;
PROCEDURE COPY_OLD_FILE;
VAR
 FROM_FILE, TO_FILE : FILE; NUM_READ, NUM_WRITTEN : WORD;
BEGIN
 ASSIGN(FROM_FILE, FILE_NAME); RESET(FROM_FILE, 1);
 ASSIGN(TO_FILE, F_NAME); REWRITE(TO_FILE, 1);
 REPEAT
  BLOCKREAD(FROM_FILE, BUF, SIZEOF(BUF), NUM_READ);
  BLOCKWRITE(TO_FILE, BUF, NUM_READ, NUM_WRITTEN);
 UNTIL ((NUM_READ = 0) OR (NUM_WRITTEN <> NUM_READ)); TEXTCOLOR(CYAN);
 WRITELN('Original file contents... ', FILESIZE(FROM_FILE),' Bytes copied in...');
 WRITELN(F_NAME); CLOSE(FROM_FILE); CLOSE(TO_FILE); TEXTCOLOR(YELLOW);
 WRITELN; WRITE('Hit any key to continue ');
 SOUND1; CH := READKEY; WRITELN;
END;
```

```
PROCEDURE FILE_NOT_DISTURBED;
BEGIN
 WRITELN;
 WRITE('Old contents of the file are not disturbed...',
       'Hit any key to check ');
 CH := READKEY; WRITELN;
END;
PROCEDURE WRITE_LINES_IN_FILE;
VAR
 F1 : FILE; C_W : INTEGER; { C_W = CANNOT WRITE }
BEGIN
 LIST_LINES(NUMBER, STOP); WRITELN; C_W := 0;
 WRITELN('Space required on disk = ', TOT_CH, ' Bytes'); WRITELN;
 ASSIGN(F1, FILE_NAME); FINDFIRST(FILE_NAME, FAANYFILE, DIRINFO);
 TEXTCOLOR(CYAN);
 IF (DOSERROR = 0) THEN
 BEGIN
  GETFATTR(F1, ATTR);
  IF (DOSERROR <> 0) THEN
  BEGIN
   WRITELN('Error code = ', DOSERROR, ' Cannot write in the file...');
   EXIT;
  END
  ELSE
  BEGIN
   IF (ATTR AND FAREADONLY  <> 0) THEN BEGIN WRITE('<Read Only>'); C_W := 1; END;
   IF (ATTR AND FAHIDDEN    <> 0) THEN BEGIN WRITE('<Hidden>');    C_W := 1; END;
   IF (ATTR AND FASYSFILE   <> 0) THEN BEGIN WRITE('<System>');    C_W := 1; END;
   IF (ATTR AND FADIRECTORY <> 0) THEN BEGIN WRITE('<Directory>'); C_W := 1; END;
   IF (ATTR AND FAVOLUMEID  <> 0) THEN BEGIN WRITE('<Volume Id>'); C_W := 1; END;
   IF (C_W = 1) THEN
   BEGIN
    TEXTCOLOR(YELLOW); WRITE(' file...Cannot write...Hit any key to continue ');
    SOUND1; CH := READKEY; WRITELN; EXIT;
   END;
  END;
 END;
 TEXTCOLOR(YELLOW); ASSIGN(F, FILE_NAME);
 {$I-} RESET(F); CLOSE(F); {$I+}
 IF (IORESULT <> 0) THEN
 BEGIN
  {$I-} REWRITE(F); CLOSE(F); {$I+}
  WRITELN('File...', FILE_NAME);
  WRITELN('...is not found...New file with size = 0 is created');
 END
 ELSE
 BEGIN
  FINDFIRST(FILE_NAME, FAANYFILE, DIRINFO);
  WRITELN('File...', FILE_NAME, ', Size...', DIRINFO.SIZE, ' Bytes');
  WRITE('Already exists...Overwrite it(old contents will be replaced) ? ',
        '(Y/y for yes) ');
  SOUND1; CH := UPCASE(READKEY); WRITELN(CH);
  IF (CH <> 'Y') THEN BEGIN FILE_NOT_DISTURBED; EXIT; END;
  WRITELN; FILESPLIT(FILE_NAME, D1, N1, E1);
  STRECOPY(STRECOPY(STRECOPY(F_NAME, D1), N1), '.OLD'); COPY_OLD_FILE;
 END;
```

```
 DIS_FILE_NAMES; WRITELN; WRITE('Do you surely want to write ? (Y/y for yes) ');
 SOUND1; CH := UPCASE(READKEY); WRITELN(CH);
 IF (CH <> 'Y') THEN BEGIN FILE_NOT_DISTURBED; EXIT; END;
 WRITELN; ASSIGN(F, FILE_NAME); SETTEXTBUF(F, BUF);
 {$I-} REWRITE(F); {$I+}
 IF ((IORESULT = 0) AND (TOT_CH > 0)) THEN
 BEGIN
  TOT_CH := 0; DONE := FALSE; LINK := HEAD^.NEXT; START := NUMBER;
  WHILE (LINK <> NIL) AND NOT DONE DO
  IF (LINK^.LINE_NO > STOP) THEN DONE := TRUE
  ELSE IF (LINK^.LINE_NO >= START) THEN
  BEGIN
   TEMP := LINK^.LINE_TEXT; GET_NO_OF_CHARS;
   FOR NU := 1 TO I DO BEGIN WRITE(F, TEMP[NU]); TOT_CH := TOT_CH + 1; END;
   LINK := LINK^.NEXT;
  END ELSE LINK := LINK^.NEXT;
  CLOSE(F); WRITELN; TEXTCOLOR(CYAN);
  WRITELN('Total lines written = ', STOP - START + 1,
          ', Total characters written = ', TOT_CH);
  WRITELN; TEXTCOLOR(YELLOW); WRITE('Writing over...Hit any key to check ');
  SOUND1; CH := READKEY; WRITELN;
 END ELSE
 BEGIN WRITELN; WRITELN('Total characters = ', TOT_CH, ', Not written'); END;
END;
PROCEDURE INSERT_LINE(NUMBER : INTEGER; VAR NEW_LINE : TEXT1);
BEGIN { INSERT LINE IN MAIN LINKED LIST }
 AVAILABLE_MEMORY := TRUE;
 IF (MEMAVAIL < 268) THEN
 BEGIN
  TEXTCOLOR(CYAN); WRITELN;
  WRITELN('Insufficient memory to insert line in main linked list...');
  WRITELN; WRITE('Hit any key to continue ');
  SOUND1; CH:=READKEY; TEXTCOLOR(YELLOW); WRITELN; AVAILABLE_MEMORY:=FALSE; EXIT;
 END;
 DONE := FALSE; PREV := HEAD; LINK := HEAD^.NEXT;
 WHILE (LINK <> NIL) AND NOT DONE DO
 IF (LINK^.LINE_NO = NUMBER) THEN
 BEGIN
  NEW(P); P^.LINE_NO := NUMBER; P^.LINE_TEXT := NEW_LINE;
  P^.NEXT := LINK; PREV^.NEXT := P;
  REPEAT
   NU := LINK^.LINE_NO; NU := NU + 1; LINK^.LINE_NO := NU; LINK := LINK^.NEXT;
  UNTIL (LINK = NIL); DONE := TRUE;
 END
 ELSE IF (LINK^.LINE_NO > NUMBER) THEN
 BEGIN
  NEW(P); P^.LINE_NO := NUMBER; P^.LINE_TEXT := NEW_LINE; P^.NEXT := LINK;
  PREV^.NEXT := P; DONE := TRUE;
 END ELSE
 BEGIN PREV := LINK; LINK := LINK^.NEXT; END;
 IF (NOT DONE) THEN
 BEGIN
  NEW(P); P^.LINE_NO := NUMBER; P^.LINE_TEXT := NEW_LINE; P^.NEXT := NIL;
  PREV^.NEXT := P;
 END;
END;
```

```
PROCEDURE INSERT_LINE1(NUMBER : INTEGER; VAR NEW_LINE1 : TEXT1);
BEGIN { INSERT LINE IN SECOND LINKED LIST FOR COPY / MOVE }
 AVAILABLE_MEMORY := TRUE;
 IF (MEMAVAIL < 268) THEN
 BEGIN
  TEXTCOLOR(CYAN); WRITELN;
  WRITELN('Insufficient memory to insert line in second linked list...');
  WRITELN; WRITE('Hit any key to continue ');
  SOUND1; CH:=READKEY; TEXTCOLOR(YELLOW); WRITELN; AVAILABLE_MEMORY:=FALSE; EXIT;
 END;
 DONE := FALSE; PREV1 := HEAD1; LINK1 := HEAD1^.NEXT1;
 WHILE (LINK1 <> NIL) AND NOT DONE DO
 IF (LINK1^.LINE_NO1 = NUMBER) THEN
 BEGIN
  NEW(P1); P1^.LINE_NO1 := NUMBER;
  P1^.LINE_TEXT1 := NEW_LINE1; P1^.NEXT1 := LINK1; PREV1^.NEXT1 := P1;
  REPEAT
   NU := LINK1^.LINE_NO1; NU := NU + 1; LINK1^.LINE_NO1 := NU;
   LINK1 := LINK1^.NEXT1;
  UNTIL (LINK1 = NIL); DONE := TRUE;
 END ELSE IF (LINK1^.LINE_NO1 > NUMBER) THEN
 BEGIN
  NEW(P1); P1^.LINE_NO1 := NUMBER; P1^.LINE_TEXT1 := NEW_LINE1;
  P1^.NEXT1 := LINK1; PREV1^.NEXT1 := P1; DONE := TRUE;
 END ELSE
 BEGIN PREV1 := LINK1; LINK1 := LINK1^.NEXT1; END;
 IF (NOT DONE) THEN
 BEGIN
  NEW(P1); P1^.LINE_NO1 := NUMBER; P1^.LINE_TEXT1 := NEW_LINE1;
  P1^.NEXT1 := NIL; PREV1^.NEXT1 := P1;
 END;
END;
PROCEDURE READ_LINES_FROM_FILE(SNO : INTEGER); { READ FROM EXTERNAL FILE }
LABEL 5;
VAR
 TEMP133, TEMP134 : CHAR;
BEGIN
 AVAILABLE_MEMORY := TRUE;
 IF (DIRINFO.SIZE <> 0) THEN
 BEGIN
  ASSIGN(F, FILE_NAME); SETTEXTBUF(F, BUF); RESET(F);
  NU := 0; TOT_CH := 0; FOR I := 1 TO 134 DO TEMP1[I] := CHR(0);
  WHILE (NOT EOF(F)) DO
  BEGIN
   READ(F, CH); IF (CH = CHR(26)) THEN GOTO 5;
   NU := NU + 1; TOT_CH := TOT_CH + 1; TEMP[NU] := CH;
   IF ((TEMP[NU] = CHR(10)) OR (NU = 134)) THEN
   BEGIN
    TEMP1[NU] := CH;
    IF (MEMAVAIL < 268) THEN
    BEGIN
     TEXTCOLOR(CYAN); WRITELN;
     WRITELN('Insufficient memory to load/copy file...Use other editor.');
     WRITELN; WRITE('Hit any key to continue '); SOUND1; CH := READKEY;
     AVAILABLE_MEMORY := FALSE; TEXTCOLOR(YELLOW); WRITELN; CLOSE(F); EXIT;
    END;
```

```
    IF ((NU = 134) AND (TEMP[133] <> CHR(13)) AND (TEMP[134] <> CHR(10))) THEN
    BEGIN
     TEMP133 := TEMP[133]; TEMP134 := TEMP[134]; TEMP1[133] := CHR(13);
     TEMP1[134] := CHR(10); NEW_LINE := TEMP1; INSERT_LINE(SNO, NEW_LINE);
     SNO := SNO + 1; TEMP1[1] := TEMP133; FOR K := 3 TO 134 DO TEMP1[K] := CHR(0);
     TEMP1[2] := TEMP134; TOT_CH := TOT_CH + 2; NU := 2;
     HIGHEST_NO := HIGHEST_NO + 1;
    END
    ELSE
    BEGIN
     NEW_LINE := TEMP1; INSERT_LINE(SNO, NEW_LINE);
     SNO := SNO + 1; HIGHEST_NO := HIGHEST_NO + 1; NU := 0;
     FOR K := 1 TO 134 DO TEMP1[K] := CHR(0);
    END
   END
   ELSE TEMP1[NU] := TEMP[NU];
  END;
5:IF (NU <> 0) THEN
  BEGIN
   TEMP1[NU+1] := CHR(13); TEMP1[NU+2] := CHR(10); TOT_CH := TOT_CH + 2;
   NEW_LINE := TEMP1; INSERT_LINE(SNO, NEW_LINE);
   SNO := SNO + 1; HIGHEST_NO := HIGHEST_NO + 1;
  END;
  WRITELN; CLOSE(F);
 END;
END;
PROCEDURE DELETE_LINE(NUMBER : INTEGER);
BEGIN
 PREV := HEAD; LINK := HEAD^.NEXT; DONE := FALSE;
 WHILE (LINK <> NIL) AND NOT DONE DO
 IF (LINK^.LINE_NO = NUMBER) THEN
 BEGIN PREV^.NEXT := LINK^.NEXT; DISPOSE(LINK); DONE := TRUE; END
 ELSE BEGIN PREV := LINK; LINK := LINK^.NEXT; END;
 LINK := PREV^.NEXT;
 IF (OPTION = 'U') THEN
 WHILE (LINK <> NIL) DO
 BEGIN
  NU := LINK^.LINE_NO; NU := NU - 1; LINK^.LINE_NO := NU; LINK := LINK^.NEXT;
 END;
END;
PROCEDURE COPY_LINES(START, STOP : INTEGER); { SECOND LINKED LIST IS USED }
LABEL 5;
BEGIN
 NEW(HEAD1); HEAD1^.NEXT1 := NIL; HIGHEST_NO1 := 1; DONE := FALSE;
 LINK := HEAD^.NEXT; I := 0;
 WHILE (LINK <> NIL) AND NOT DONE DO
 IF (LINK^.LINE_NO > STOP) THEN DONE := TRUE
 ELSE IF (LINK^.LINE_NO >= START) THEN
 BEGIN
  TEMP := LINK^.LINE_TEXT; NEW_LINE := TEMP; I := I + 1;
  NEW_LINE1 := NEW_LINE; INSERT_LINE1(HIGHEST_NO1, NEW_LINE1);
  IF (NOT(AVAILABLE_MEMORY)) THEN GOTO 5;
  HIGHEST_NO1 := HIGHEST_NO1 + 1; LINK := LINK^.NEXT;
 END
 ELSE LINK := LINK^.NEXT;
5:END;
```

```
PROCEDURE CHECK_IGNORE_CASE;
BEGIN
 FOR J := 0 TO L - 1 DO
 BEGIN
  TEMP14[J+1] := TEMP[I1+J];
  IF (IGNORE_CASE = 'Y') THEN
  IF ((ORD(TEMP14[J+1]) >= 97) AND (ORD(TEMP14[J+1]) <= 122)) THEN
  TEMP14[J+1] := CHR(ORD(TEMP14[J+1]) - 32);
 END;
END;
PROCEDURE SEARCH_STRING(START, STOP : INTEGER; VAR TEMP12 : STRING12);
LABEL 1;
BEGIN
 LINE_OCC := 0; TOT_OCC := 0; K1 := 0; DONE := FALSE; LINK := HEAD^.NEXT;
 WHILE (LINK <> NIL) AND NOT DONE DO
 BEGIN
  IF (LINK^.LINE_NO > STOP) THEN DONE := TRUE
  ELSE IF (LINK^.LINE_NO >= START) THEN
  BEGIN
   TEMP := LINK^.LINE_TEXT; GET_NO_OF_CHARS; I1 := 1;
   WHILE (I1 <= I) DO
   BEGIN
    CHECK_IGNORE_CASE; { PUT INPUT STRING LENGTH CHARACTERS INTO TEMP14 }
    IF (WILD = 'Y') THEN
    FOR J := 1 TO L DO IF (TEMP12[J] = '?') THEN TEMP14[J] := '?';
    IF (TEMP14 <> TEMP12) THEN I1 := I1 + 1
    ELSE
    BEGIN
     LINE_OCC := LINE_OCC + 1; TOT_OCC := TOT_OCC + 1; I1 := I1 + L;
    END;
   END;
   IF (LINE_OCC > 0) THEN
   BEGIN
    K1 := K1 + 1; WRITE(LINK^.LINE_NO:4, ':'); J := 1;
    WHILE (TEMP[J] <> CHR(13)) DO BEGIN WRITE(TEMP[J]); J := J + 1; END;
    WRITELN; TEXTCOLOR(CYAN);
    WRITELN('Number of occurrences in this line = ', LINE_OCC);
    TEXTCOLOR(YELLOW); K := K1 DIV 7;
    IF (K1 = K * 7) THEN
    BEGIN
     WRITELN;
     WRITE('Seven lines displayed...Enter S/s to stop or Hit any key to
            continue');
     SOUND1; CH := UPCASE(READKEY);
     IF (CH = 'S') THEN BEGIN WRITELN; GOTO 1; END;
     WRITELN(CLEAR1);
    END;
   END;
  END;
  LINK := LINK^.NEXT; LINE_OCC := 0;
 END;
1:WRITELN;
 WRITELN('Total occurrences = ', TOT_OCC);
 WRITELN; WRITE('Search over...Hit any key to continue ');
 SOUND1; CH := READKEY; WRITELN;
END;
```

```
PROCEDURE REPLACE_STRING(START, STOP : INTEGER; VAR TEMP12, TEMP13 : STRING12);
LABEL 32, 50, 51, 52;
BEGIN
 LINK := HEAD^.NEXT; OCC := 0; TOT_OCC := 0;
 REP_DONE := 0; LINE_REP := 0; LINE_OCC := 0; LINE_OCC1 := 0;
 DONE := FALSE; OVER_FLOW := 'N';
 IF (QUERY = 'Y') THEN WRITELN;
 WHILE (LINK <> NIL) AND NOT DONE DO
 IF (LINK^.LINE_NO > STOP) THEN DONE := TRUE
 ELSE IF (LINK^.LINE_NO >= START) THEN
 BEGIN
  TEMP := LINK^.LINE_TEXT; TEMP2 := LINK^.LINE_TEXT;
  GET_NO_OF_CHARS; I1 := 1; T_CH := I;
  WHILE (I1 <= I) DO
  BEGIN
   CHECK_IGNORE_CASE; { PUT INPUT STRING LENGTH CHARACTERS INTO TEMP14 }
   IF (WILD = 'Y') THEN
   FOR J := 1 TO L DO IF (TEMP12[J] = '?') THEN TEMP14[J] := '?';
   IF (TEMP14 <> TEMP12) THEN BEGIN I1 := I1 + 1; GOTO 32; END;
   OCC := OCC + 1; LINE_OCC := LINE_OCC + 1;
   LINE_OCC1 := LINE_OCC1 + 1; TOT_OCC := TOT_OCC + 1;
   IF ((T_CH + (M - L) > 132) AND (M > L)) THEN OVER_FLOW := 'Y';
   IF ((QUERY = 'Y') OR (OVER_FLOW = 'Y')) THEN
   BEGIN
    WRITE(LINK^.LINE_NO:4, ':'); NU := 1;
    REPEAT WRITE(TEMP2[NU]); NU := NU + 1;
    UNTIL ((TEMP2[NU] = CHR(13)) OR (NU = 133));
    WRITELN;
    IF (OVER_FLOW = 'Y') THEN
    BEGIN
     TEXTCOLOR(CYAN);
     WRITELN('Line over flow...Last ', M - L,
             ' Character(s) will be lost...');
     TEXTCOLOR(YELLOW);
    END;
    WRITE('Original line occurrence = ', LINE_OCC1);
    TEXTCOLOR(CYAN);
    WRITE(', Replace ? (Y/y for yes...A/a to abort) ');
    TEXTCOLOR(YELLOW); SOUND1; CH := UPCASE(READKEY);
    WRITELN(CH); WRITELN;
    IF (CH = 'Y') THEN GOTO 50;
    IF (CH = 'A') THEN GOTO 51;
    I1 := I1 + L; OCC := OCC - 1; LINE_OCC := LINE_OCC - 1; GOTO 32;
   END;
50:IF (L = M) THEN FOR J := 0 TO L - 1 DO TEMP2[I1+J] := TEMP13[J+1]
   ELSE IF (L < M) THEN
   BEGIN
    J1 := I1 + (OCC - 1) * (M - L);
    FOR J := 0 TO L - 1 DO TEMP2[J1+J] := TEMP13[J+1]; J := J1 + L;
    FOR K := I + (M - L) * (LINE_OCC - 1) DOWNTO J
    DO TEMP2[K+M-L] := TEMP2[K];
    J := J1; FOR K := L + 1 TO M DO
    BEGIN TEMP2[J+L] := TEMP13[K]; J := J + 1; END;
    T_CH := T_CH + (M - L); IF (OVER_FLOW = 'Y') THEN TEMP2[133] := #0;
   END
   ELSE
```

```
   BEGIN
    J1 := I1 - (OCC - 1) * (L - M);
    FOR J := 0 TO M - 1 DO TEMP2[J1+J] := TEMP13[J+1];
    J := (I1 + L) - (L - M) * OCC;
    FOR K := I1 + L TO I DO BEGIN TEMP2[J] := TEMP[K]; J := J + 1; END;
    FOR K := I DOWNTO I - LINE_OCC * (L - M) + 1 DO TEMP2[K] := CHR(0);
   END;
   I1 := I1 + L; REP_DONE := REP_DONE + 1; LINE_REP := LINE_REP + 1; SAVED:=FALSE;
32:END;
51:IF (LINE_REP > 0) THEN
   BEGIN
    IF ((QUERY = 'Y') OR (OVER_FLOW = 'Y')) THEN
    BEGIN
     WRITE(LINK^.LINE_NO:4, ':'); NU := 1;
     REPEAT WRITE(TEMP2[NU]); NU := NU + 1;
     UNTIL ((TEMP2[NU] = CHR(13)) OR (NU = 133)); WRITELN; WRITELN;
    END;
    I := 135;
    REPEAT I := I - 1; UNTIL (TEMP2[I] <> CHR(0));
    IF ((TEMP2[I-1] <> CHR(13)) AND (TEMP2[I] <> CHR(10))) THEN
    BEGIN
     TEMP2[I+1] := CHR(13); TEMP2[I+2] := CHR(10);
     IF (I >= 133) THEN
     BEGIN TEMP2[133] := CHR(13); TEMP2[134] := CHR(10); END;
     LINK^.LINE_TEXT := TEMP2;
    END
    ELSE LINK^.LINE_TEXT := TEMP2;
   END;
   IF (((QUERY = 'Y') OR (OVER_FLOW = 'Y')) AND (CH = 'A')) THEN GOTO 52;
   OVER_FLOW := 'N'; OCC := 0; LINE_OCC := 0; LINE_OCC1 := 0; LINE_REP := 0;
   LINK := LINK^.NEXT;
  END
  ELSE LINK := LINK^.NEXT;
52:WRITELN;
 WRITELN('Total occurrences = ', TOT_OCC);
 WRITELN; WRITELN('Total replacements done = ', REP_DONE);
 WRITELN; WRITE('Replacement over...Hit any key to continue ');
 SOUND1; CH := READKEY; WRITELN;
END;
PROCEDURE UPPER_CASE(START, STOP : INTEGER);
BEGIN
 DONE := FALSE; LINK := HEAD^.NEXT;
 WHILE (LINK <> NIL) AND NOT DONE DO
 IF (LINK^.LINE_NO > STOP) THEN DONE := TRUE
 ELSE IF (LINK^.LINE_NO >= START) THEN
 BEGIN
  TEMP := LINK^.LINE_TEXT; GET_NO_OF_CHARS;
  FOR I1 := 1 TO I DO
  BEGIN
   CH := TEMP[I1];
   IF ((ORD(CH) >= 97) AND (ORD(CH) <= 122)) THEN TEMP[I1] := CHR(ORD(CH) - 32);
  END;
  LINK^.LINE_TEXT := TEMP; LINK := LINK^.NEXT;
 END
 ELSE LINK := LINK^.NEXT;
END;
```

```
PROCEDURE LOWER_CASE(START, STOP : INTEGER);
BEGIN
 DONE := FALSE; LINK := HEAD^.NEXT;
 WHILE (LINK <> NIL) AND NOT DONE DO
 IF (LINK^.LINE_NO > STOP) THEN DONE := TRUE
 ELSE IF (LINK^.LINE_NO >= START) THEN
 BEGIN
  TEMP := LINK^.LINE_TEXT; GET_NO_OF_CHARS;
  FOR I1 := 1 TO I DO
  BEGIN
   CH := TEMP[I1];
   IF ((ORD(CH) >= 65) AND (ORD(CH) <= 90)) THEN
   TEMP[I1] := CHR(ORD(CH) + 32);
  END;
  LINK^.LINE_TEXT := TEMP; LINK := LINK^.NEXT;
 END
 ELSE LINK := LINK^.NEXT
END;
PROCEDURE TITLE_SENTENCE_CASE(START, STOP : INTEGER);
BEGIN
 DONE := FALSE; LINK := HEAD^.NEXT;
 WHILE (LINK <> NIL) AND NOT DONE DO
 IF (LINK^.LINE_NO > STOP) THEN DONE := TRUE
 ELSE IF (LINK^.LINE_NO >= START) THEN
 BEGIN
  TEMP := LINK^.LINE_TEXT; GET_NO_OF_CHARS;
  TEMP[1] := UPCASE(TEMP[1]); OLD_CH := TEMP[1];
  FOR I1 := 2 TO I DO
  BEGIN
   CH := TEMP[I1];
   IF ((OLD_CH IN WORD_SEPARATORS) AND (CHOICE = 'T')) THEN
   TEMP[I1] := UPCASE(TEMP[I1]);
   IF ((OLD_CH IN SENTENCE_SEPARATORS) AND (CHOICE = 'S')) THEN
   BEGIN TEMP[I1] := UPCASE(TEMP[I1]); SEN_SEP := TRUE; END;
   IF ((CH <> ' ') AND (SEN_SEP = TRUE)) THEN
   BEGIN TEMP[I1] := UPCASE(TEMP[I1]); SEN_SEP := FALSE; END;
   OLD_CH := CH;
  END;
  LINK^.LINE_TEXT := TEMP; LINK := LINK^.NEXT;
 END
 ELSE LINK := LINK^.NEXT
END;
PROCEDURE DATE_TIME;
BEGIN
 GETDATE(YEAR, MONTH, DAY, DAYOFWEEK);
 DAY_NAM := DAY_NAME[DAYOFWEEK];
 GETTIME(HOURS, MINUTES, SECONDS, HUNDREDTHS);
END;
PROCEDURE PAGE_HEADING(PGNO : INTEGER);
BEGIN
 DATE_TIME; WRITELN(PRINTER);
 WRITELN(PRINTER, 'File:', FILE_NAME, ' Date:', DAY, '/', MONTH, '/', YEAR,
         ' Day:', DAY_NAM, ' Time:', HOURS, ':', MINUTES, ':', SECONDS,
         ' Pg.No.:', PGNO);
 WRITELN(PRINTER); LINENUM := LINENUM + (PAGE_HEADING_LINES + 2);
END;
```

```
PROCEDURE PRINT_FILE;
LABEL 5, 6;
VAR
 CH1 : CHAR;
BEGIN
 WRITELN('Printing a file on parallel printer i.e. LPT1/PRN');
 OLD_FILE_NAME := FILE_NAME; GET_CHECK_FILE_NAME1;
 STRECOPY(STRECOPY(ASCIIZ, FILE_NAME), CHR(0));
 ASSIGN(F, ASCIIZ); SETTEXTBUF(F, BUF);
 {$I-} RESET(F); {$I+}
 IF (IORESULT <> 0) THEN
 BEGIN
  TEXTCOLOR(CYAN); WRITELN('File does not exist/Wrong path so no printing');
  TEXTCOLOR(YELLOW); GOTO 6;
 END;
 CLOSE(F);
 REPEAT
  TEXTCOLOR(CYAN);
  WRITELN('Please save your work before proceeding...Runtime error may occur...');
  WRITE('Choose...1..To print or 2..To abort ');
  SOUND1; CH := READKEY; WRITELN(CH);
  TEXTCOLOR(YELLOW); VAL(CH, I, CODE);
  IF ((I < 1) OR (I > 2)) THEN
  BEGIN
   WRITELN; WRITELN('Invalid choice...Re-enter'); WRITELN;
  END;
 UNTIL ((I = 1) OR (I = 2));
 IF (I = 2) THEN GOTO 6;
 REPEAT
  WRITELN; SOUND1;
  WRITE('Choose...1..For 80 column printer or 2..For 132 column printer ');
  CH := READKEY; WRITELN(CH); VAL(CH, J, CODE);
  IF ((J < 1) OR (J > 2)) THEN
  BEGIN
   WRITELN; TEXTCOLOR(CYAN); WRITELN('Invalid choice...Re-enter');
   TEXTCOLOR(YELLOW);
  END;
 UNTIL ((J = 1) OR (J = 2));
 ASSIGN(PRINTER, 'LPT1'); REWRITE(PRINTER); LINENUM := 0; I := 0;
 ASSIGN(F, ASCIIZ); SETTEXTBUF(F, BUF); {$I-} RESET(F); {$I+}
 WRITELN; WRITE('Do you want page heading ? (Y/y for yes) ');
 SOUND1; CH1 := READKEY; WRITELN(CH1); IF (CH1 = CHR(0)) THEN CH1 := 'N';
 IF (UPCASE(CH1) = 'Y') THEN
 BEGIN
  L := 0; WHILE (FILE_NAME[L] <> #0) DO L := L + 1;
  PAGE_HEADING_CHARS := L + 56;
  IF (J = 1) THEN
  IF (PAGE_HEADING_CHARS <= 80) THEN PAGE_HEADING_LINES := 1
  ELSE PAGE_HEADING_LINES := 2
  ELSE
  IF (PAGE_HEADING_CHARS <= 132) THEN PAGE_HEADING_LINES := 1
  ELSE PAGE_HEADING_LINES := 2;
  PGNO := 1; PAGE_HEADING(PGNO);
 END
 ELSE
 BEGIN LINENUM := LINENUM + 3; FOR K := 1 TO 3 DO WRITELN(PRINTER); END;
```

```
 WHILE (NOT EOF(F)) DO
 BEGIN
  IF (J = 1) THEN
  REPEAT
   READ(F, CH); I := I + 1; IF (CH = CHR(26)) THEN GOTO 5;
   WRITE(PRINTER, CH);
  UNTIL ((CH = CHR(10)) OR (I = 82))
  ELSE
  REPEAT
   READ(F, CH); I := I + 1; IF (CH = CHR(26)) THEN GOTO 5;
   WRITE(PRINTER, CH);
  UNTIL ((CH = CHR(10)) OR (I = 134));
  LINENUM := LINENUM + 1; I := 0;
  IF (LINENUM = 69) THEN
  BEGIN
   LINENUM := 0; FOR K := 1 TO 3 DO WRITELN(PRINTER);
   IF (UPCASE(CH1) = 'Y') THEN
   BEGIN PGNO := PGNO + 1; PAGE_HEADING(PGNO); END
   ELSE
   BEGIN LINENUM := LINENUM + 3; FOR K := 1 TO 3 DO WRITELN(PRINTER); END;
  END;
 END;
5:WRITELN; WRITE(PRINTER, CHR(12));
 WRITELN('Printing over...If machine hangs, press Alt & Enter, ',
         'Choose close from title bar'); CLOSE(PRINTER); CLOSE(F);
6:WRITELN; WRITE('Hit any key to continue ');
 SOUND1; CH := READKEY; WRITELN; FILE_NAME := OLD_FILE_NAME;
END;
PROCEDURE APPEND_INSERT_UPDATE_MESSAGE;
BEGIN
 WRITELN('Use F1 or --> to take a character from the  previous  line ',
         'in the current line.');
 WRITELN('Use F2 or <-- or Del to delete a character from previous line ',
         'for current line.');
 WRITELN('(F2 or <-- or Del  does not  physically  delete a character ',
         'from previous line)');
 WRITELN('Use F3 or End to take  all  the  remaining  characters ',
         'from the previous line.');
 WRITELN('Tab key inserts 5 blank characters. ',
         'Backspace key  deletes previous character.');
END;
PROCEDURE SET_FILE_ATTRIBUTES;
LABEL 7, 60;
CONST
 CHECK_ATTRIBUTES = $43; GET_ATTRIBUTES = 0; SET_ATTRIBUTES = 1; CARRY_FLAG = 1;
VAR
 REG : TREGISTERS; RESFILE : FILE; CH1 : CHAR;
 ATTRIBUTES, MASK, I, PX, ABC, P, Q : INTEGER;
 T : ARRAY [1..6] OF INTEGER;
BEGIN
 WITH REG DO
 BEGIN
  OLD_FILE_NAME := FILE_NAME; GET_CHECK_FILE_NAME1;
  STRECOPY(STRECOPY(FILE_NAME, ASCIIZ), CHR(0)); TEXTCOLOR(CYAN);
  AH := CHECK_ATTRIBUTES; DS := SEG(FILE_NAME);
  DX := OFS(FILE_NAME); AL := GET_ATTRIBUTES; MSDOS(REG);
```

```
 IF ((FLAGS AND CARRY_FLAG) = 1) THEN
 BEGIN
  WRITELN; WRITELN('Unable to find...', FILE_NAME, '...Job aborted');
  TEXTCOLOR(YELLOW); GOTO 7;
 END ELSE
 BEGIN
  ATTRIBUTES := CX; PX := REG.CX; WRITELN;
  WRITELN('File attributes of...', FILE_NAME, '...are');
  IF (ATTRIBUTES = 0) THEN WRITE('<Normal Read/Write>')
  ELSE
  BEGIN
   MASK := 1;
   FOR I := 1 TO 6 DO
   BEGIN
    CX := ATTRIBUTES AND MASK;
    IF (CX = 1)  THEN WRITE('<Read Only>');
    IF (CX = 2)  THEN WRITE('<Hidden>');
    IF (CX = 4)  THEN WRITE('<System>');
    IF (CX = 8)  THEN WRITE('<Volume Id>');
    IF (CX = 16) THEN WRITE('<Directory>');
    IF (CX = 32) THEN WRITE('<Archive>');
    MASK := MASK * 2;
   END;
  END;
 END;
END;
TEXTCOLOR(YELLOW); WRITELN; WRITELN;
WRITE('Do you want to change the attributes ? (Y/y for yes) ');
SOUND1; CH := READKEY; WRITELN(CH);
IF (UPCASE(CH) = 'Y') THEN
BEGIN
 WRITELN; REG.AH := CHECK_ATTRIBUTES; REG.DS := SEG(FILE_NAME);
 REG.DX := OFS(FILE_NAME); REG.AL := SET_ATTRIBUTES; GOTO 60;
END;
WRITELN; WRITELN('Attributes are not disturbed...'); GOTO 7;
60:WRITELN; WRITELN('Change attributes...');
FOR P := 1 TO 6 DO T[P] := 0;
ABC := PX; P := 1; IF (ABC = 0) THEN FOR P := 1 TO 6 DO T[P] := 0;
WHILE (ABC <> 0) DO
BEGIN T[P] := ABC MOD 2; ABC := ABC DIV 2; P := P + 1; END;
IF ((T[5] = 1) OR (T[4] = 1)) THEN
BEGIN
 WRITELN; TEXTCOLOR(CYAN); WRITELN('Cannot change attribute...');
 TEXTCOLOR(YELLOW); GOTO 7;
END;
IF (T[6] = 1) THEN
BEGIN
 WRITELN; WRITELN('The file is Archive file');
 WRITE('Do you want to make it Non Archive ? (Y/y for yes) ');
 SOUND1; CH := UPCASE(READKEY); WRITELN(CH); IF (CH = 'Y') THEN T[6] := 0;
END ELSE
BEGIN
 WRITELN; WRITELN('The file is Non Archive file');
 WRITE('Do you want to make it Archive ? (Y/y for yes) ');
 SOUND1; CH := UPCASE(READKEY); WRITELN(CH); IF (CH = 'Y') THEN T[6] := 1;
END;
```

```
 IF (T[3] = 1) THEN
 BEGIN
  WRITELN; WRITELN('The file is System file');
  WRITE('Do you want to make it Non System file ? (Y/y for yes) ');
  SOUND1; CH := UPCASE(READKEY); WRITELN(CH); IF (CH = 'Y') THEN T[3] := 0;
 END
 ELSE
 BEGIN
  WRITELN; WRITELN('The file is Non System file');
  WRITE('Do you want to make it System file ? (Y/y for yes) ');
  SOUND1; CH := UPCASE(READKEY); WRITELN(CH); IF (CH = 'Y') THEN T[3] := 1;
 END;
 IF (T[2] = 1) THEN
 BEGIN
  WRITELN; WRITELN('The file is Hidden file');
  WRITE('Do you want to make it Non Hidden ? (Y/y for yes) ');
  SOUND1; CH := UPCASE(READKEY); WRITELN(CH); IF (CH = 'Y') THEN T[2] := 0;
 END
 ELSE
 BEGIN
  WRITELN; WRITELN('The file is Non Hidden file');
  WRITE('Do you want to make it Hidden ? (Y/y for yes) ');
  SOUND1; CH := UPCASE(READKEY); WRITELN(CH); IF (CH = 'Y') THEN T[2] := 1;
 END;
 IF (T[1] = 1) THEN
 BEGIN
  WRITELN; WRITELN('The file is Read Only file');
  WRITE('Do you want to make it Non Read Only ? (Y/y for yes) ');
  SOUND1; CH := UPCASE(READKEY); WRITELN(CH); IF (CH = 'Y') THEN T[1] := 0;
 END
 ELSE
 BEGIN
  WRITELN; WRITELN('The file is Non Read Only file');
  WRITE('Do you want to make it Read Only ? (Y/y for yes) ');
  SOUND1; CH := UPCASE(READKEY); WRITELN(CH); IF (CH = 'Y') THEN T[1] := 1;
 END;
 REG.CX := T[6] * 32 + T[5] * 16 + T[4] * 8 + T[3] * 4 + T[2] * 2 + T[1] * 1;
 MSDOS(REG); WRITELN; WRITELN;
 WRITELN('The attribute(s) of file...', FILE_NAME, '...are set as');
 F_NAME := FILE_NAME; DISPLAY_FILE_ATTRIBUTES;
7:WRITELN; WRITE('Hit any key to recheck '); SOUND1; CH := READKEY; WRITELN;
 DISPLAY_DIRECTORY; FILE_NAME := OLD_FILE_NAME;
END;
PROCEDURE SEARCH_FILES;
LABEL 10, 12;
VAR
 FILEVAR : TEXT; COUNT, COUNT1, COUNT2, LINENUMBER : INTEGER;
 ONELINE, SUBSTRING, ONELINE_OLD, SUBSTRING_OLD : STRING; FOUND : BOOLEAN;
BEGIN
 GET_CHECK_FILE_NAME2; COUNT := 0; COUNT1 := 0; COUNT2 := 0; WRITELN;
 WRITE('Search for(Maximum 255 characters) ? (Hit just Enter to quit) ');
 SOUND1; READLN(SUBSTRING);
 IF (LENGTH(SUBSTRING) = 0) THEN BEGIN WRITELN; GOTO 10; END;
 SUBSTRING_OLD := SUBSTRING; TEXTCOLOR(CYAN); WRITELN;
 WRITE('Do you want to ignore case ? (Y/y for yes) '); SOUND1;
 IGNORE_CASE := UPCASE(READKEY); WRITELN(IGNORE_CASE); WRITELN; TEXTCOLOR(YELLOW);
```

```
 IF (IGNORE_CASE = 'Y') THEN
 FOR I := 1 TO LENGTH(SUBSTRING) DO SUBSTRING[I] := UPCASE(SUBSTRING[I]);
 WRITELN('Searching...', ASCIIZ, ' for...', SUBSTRING_OLD);
 WRITELN; FILESPLIT(ASCIIZ, D1, N1, E1); FINDFIRST(ASCIIZ, FAARCHIVE, DIRINFO);
 IF (DOSERROR <> 0) THEN GOTO 12; FOUND := FALSE;
 WHILE (DOSERROR = 0) DO
 BEGIN
  COUNT := COUNT + 1;
  WRITELN('Searching file number...', COUNT:5, ', Name = ', DIRINFO.NAME:12,
          ', Size =', DIRINFO.SIZE:10, ' Bytes');
  STRECOPY(STRECOPY(F_NAME, D1), DIRINFO.NAME);
  ASSIGN(FILEVAR, F_NAME); RESET(FILEVAR); LINENUMBER := 0;
  WHILE (NOT EOF(FILEVAR)) DO
  BEGIN
   READLN(FILEVAR, ONELINE); LINENUMBER := LINENUMBER + 1; ONELINE_OLD := ONELINE;
   IF (IGNORE_CASE = 'Y') THEN
   FOR I := 1 TO LENGTH(ONELINE) DO ONELINE[I] := UPCASE(ONELINE[I]);
   IF (POS(SUBSTRING, ONELINE) > 0) THEN
   BEGIN
    WRITELN(LINENUMBER:5, ':', ONELINE_OLD); COUNT1 := COUNT1 + 1; FOUND := TRUE;
    K := COUNT1 DIV 8;
    IF (COUNT1 = K * 8) THEN
    BEGIN
     WRITE('Eight lines displayed...Hit any key ');
     SOUND1; CH := READKEY; WRITELN;
    END;
   END;
  END;
  IF (FOUND) THEN
  BEGIN
   TEXTCOLOR(CYAN);
   WRITELN('Number of lines with the string in the file = ', COUNT1);
   WRITELN; WRITE('Enter S/s to stop or other character to continue ');
   TEXTCOLOR(YELLOW); SOUND1; CH := UPCASE(READKEY);
   IF (CH = 'S') THEN BEGIN WRITELN; GOTO 12; END;
   WRITELN(CLEAR1); COUNT1 := 0; COUNT2 := COUNT2 + 1; FOUND := FALSE;
  END;
  CLOSE(FILEVAR); FINDNEXT(DIRINFO);
 END;
12:WRITELN; WRITELN('Number of files searched = ', COUNT);
 WRITELN; WRITELN('Number of files having the string = ', COUNT2);
 WRITELN; WRITELN('File(s) search over ...');
10:WRITELN; WRITE('Hit any key to continue '); SOUND1; CH := READKEY; WRITELN;
END;
PROCEDURE DELETE_FILES;
LABEL 10, 20;
VAR
 F : FILE;
PROCEDURE ERASE_FILE;
VAR C_D : INTEGER; { C_D = CANNOT DELETE }
BEGIN
 FILESPLIT(ASCIIZ, D1, N1, E1); TEXTCOLOR(CYAN); C_D := 0;
 STRECOPY(STRECOPY(F_NAME,D1), DIRINFO.NAME); ASSIGN(F,F_NAME); GETFATTR(F, ATTR);
 IF (DOSERROR <> 0) THEN
 BEGIN WRITELN('Error code = ', DOSERROR, ' Cannot delete file...'); EXIT; END
 ELSE
```

```
 BEGIN
  IF (ATTR AND FAREADONLY  <> 0) THEN BEGIN WRITE('<Read Only>'); C_D := 1; END;
  IF (ATTR AND FAHIDDEN    <> 0) THEN BEGIN WRITE('<Hidden>');    C_D := 1; END;
  IF (ATTR AND FASYSFILE   <> 0) THEN BEGIN WRITE('<System>');    C_D := 1; END;
  IF (ATTR AND FADIRECTORY <> 0) THEN BEGIN WRITE('<Directory>'); C_D := 1; END;
  IF (ATTR AND FAVOLUMEID  <> 0) THEN BEGIN WRITE('<Volume Id>'); C_D := 1; END;
  IF (C_D = 1) THEN
  BEGIN
   TEXTCOLOR(YELLOW); WRITELN(' file...');
   WRITE('Cannot delete...Hit any key to continue ');
   SOUND1; CH := READKEY; WRITELN; EXIT;
  END;
  IF (ATTR AND FAARCHIVE <> 0) THEN
  BEGIN
   ERASE(F); I := I + 1; WRITE('...Deleted');
  END;
  TEXTCOLOR(YELLOW);
 END;
END;
BEGIN
 GET_CHECK_FILE_NAME2; FINDFIRST(ASCIIZ, FAANYFILE, DIRINFO);
 IF (DOSERROR <> 0) THEN
 BEGIN
  TEXTCOLOR(CYAN); WRITE('No such file(s) exist...'); TEXTCOLOR(YELLOW);
  I := 0; GOTO 20;
 END;
 WRITELN('Confirm delete'); WRITE(ASCIIZ, ' ? ', '(Y/y for yes) ');
 SOUND1; CH := UPCASE(READKEY); WRITELN(CH); I := 0;
 IF (CH <> 'Y') THEN
 BEGIN WRITELN; WRITELN('Deletion aborted...'); END
 ELSE
 BEGIN
  IF ((POS('*', ASCIIZ) > 0) OR (POS('?', ASCIIZ) > 0)) THEN
  BEGIN
   TEXTCOLOR(CYAN); WRITELN; WRITE('Delete by query ? (Y/y for yes) ');
   SOUND1; CH := UPCASE(READKEY); TEXTCOLOR(YELLOW); WRITELN(CH);
   IF (CH = 'Y') THEN GOTO 10;
  END;
  FINDFIRST(ASCIIZ, FAANYFILE, DIRINFO);
  WHILE (DOSERROR = 0) DO
  BEGIN
   WRITELN; WRITE('File - ', DIRINFO.NAME:12);
   ERASE_FILE; FINDNEXT(DIRINFO);
  END;
  GOTO 20;
10:FINDFIRST(ASCIIZ, FAANYFILE, DIRINFO);
  WHILE (DOSERROR = 0) DO
  BEGIN
   WRITELN; WRITE('Delete File - ', DIRINFO.NAME:12, ' ? (Y/y for yes) ');
   SOUND1; CH := UPCASE(READKEY); WRITE(CH);
   IF (CH = 'Y') THEN ERASE_FILE; FINDNEXT(DIRINFO);
  END;
20:WRITELN; WRITELN; WRITELN('Number of files deleted = ', I);
 END;
 TEXTCOLOR(YELLOW);
END;
```

```
PROCEDURE NUMBER_CHECK(NUMBER1 : INTEGER);
BEGIN
 TEXTCOLOR(CYAN);
 IF (NUMBER1 = 0) THEN
 BEGIN
  WRITELN('Error in number, Value = 0, New value = 1'); WRITELN;
 END;
 IF (NUMBER1 < 0) THEN
 BEGIN
  WRITELN('Error in number, Value is negative, New value = 1'); WRITELN;
 END;
 TEXTCOLOR(YELLOW);
END;
PROCEDURE INSERT_REQUIRED_BLANK_LINES(NUMBER1 : INTEGER);
BEGIN
 TEMP[1] := CHR(13); TEMP[2] := CHR(10);
 FOR I := 3 TO 134 DO TEMP[I] := CHR(0);
 B := TEMP; NEW_LINE := B; L1 := HIGHEST_NO;
 FOR L := L1 TO NUMBER1 - 1 DO
 BEGIN
  INSERT_LINE(L1, NEW_LINE);
  IF (NOT(AVAILABLE_MEMORY)) THEN EXIT;
  HIGHEST_NO := HIGHEST_NO + 1; LIST_LINES(L, L);
 END;
END;
PROCEDURE GET_LINES;
LABEL 5;
BEGIN
5:WRITELN; WRITE('From which line number ? ');
 SOUND1; READLN(A);
 VAL(A, NUMBER, CODE); NUMBER_CHECK(NUMBER);
 IF (NUMBER < 1) THEN NUMBER := 1;
 WRITE('To   which line number ? ');
 SOUND1; READLN(A); VAL(A, STOP, CODE);
 NUMBER_CHECK(STOP); IF (STOP < 1) THEN STOP := 1;
 IF (STOP > HIGHEST_NO - 1) THEN STOP := HIGHEST_NO - 1;
 IF ((NUMBER > STOP) OR (NUMBER > HIGHEST_NO - 1)) THEN
 BEGIN
  TEXTCOLOR(CYAN); WRITELN;
  WRITELN('Invalid range of numbers...From > To or From > Highest number');
  WRITELN('Maximum line number existing = ', HIGHEST_NO - 1,
          '. Retry...');
  TEXTCOLOR(YELLOW);
  GOTO 5;
 END;
END;
PROCEDURE ZERO_LINES;
BEGIN
 TEXTCOLOR(CYAN); WRITELN;
 WRITELN('Maximum line number existing = 0'); WRITELN;
 TEXTCOLOR(YELLOW);
 WRITE('Hit any key to continue ');
 SOUND1; CH := READKEY; WRITELN;
END;
```

```
PROCEDURE SEARCH_OPTIONS;
BEGIN
 TEXTCOLOR(CYAN);
 WRITE('Do you want to ignore case ? (Y/y for yes) ');
 SOUND1; IGNORE_CASE := UPCASE(READKEY);
 WRITELN(IGNORE_CASE); WRITELN;
 WRITE('? can be used as a wild character ',
       'e.g. s???o shall match with satao, spqro, etc.',
       'Be careful in using it...Use it ? (Y/y for yes) ');
 SOUND1; WILD := UPCASE(READKEY); WRITELN(WILD);
 TEXTCOLOR(YELLOW); WRITELN; L := 0;
 FOR I := 1 TO 20 DO
 BEGIN TEMP12[I] := CHR(0); TEMP14[I] := CHR(0); END;
 WRITELN('Which string ? (First 20 characters are considered...',
         'Hit just Enter to abort)');
 SOUND1; WHILE (NOT EOLN) DO BEGIN L := L + 1; READ(TEMP12[L]); END;
 READLN; IF (L = 0) THEN EXIT;
 IF (L > 20) THEN
 BEGIN
  WRITELN('Number of characters > 20, First 20 characters are considered');
  L := 20;
 END;
 IF (IGNORE_CASE = 'Y') THEN { CHANGE INPUT STRING INTO CAPITAL }
 FOR I := 1 TO L DO
 BEGIN
  CH := TEMP12[I];
  IF ((ORD(CH) >= 97) AND (ORD(CH) <= 122)) THEN
  TEMP12[I] := CHR(ORD(ORD(CH) - 32));
 END;
 B12 := TEMP12; WRITELN;
END;
PROCEDURE INSERT_APPEND1;
LABEL 101, 81, 82, 83, 53;
BEGIN
101:L := L + 1; CH := READKEY;
 IF (CH = CHR(4))  THEN GOTO 83;     { EOT i.e. Control D }
 IF (CH = CHR(13)) THEN GOTO 81;     { CARRIAGE RETURN }
 IF (L = 133) THEN GOTO 82;          { LINE OVERFLOW }
 IF ((CH = CHR(8)) AND (L = 1)) THEN { BACKSPACE ON THE LINE'S FIRST CHAR }
 BEGIN
  GOTOXY(1, WHEREY); WRITE(NUMBER:4, ':'); L := 0; GOTO 101;
 END;
 IF (CH = CHR(8)) THEN               { BACKSPACE }
 BEGIN
  FOR N := L TO 134 DO TEMP[N-1] := TEMP[N]; L := L - 2;
  GOTOXY(WHEREX - 1, WHEREY); WRITE(' ');
  GOTOXY(WHEREX - 1, WHEREY);
  GOTO 101;
 END;
 IF (CH = CHR(9)) THEN               { HORIZONTAL TAB }
 BEGIN
  FOR N := 129 DOWNTO L DO TEMP[N+5] := TEMP[N];
  FOR I := L TO L + 4 DO
  BEGIN TEMP[I] := ' '; WRITE(TEMP[I]); END;
  L := L + 4; GOTO 101;
 END;
```

```
 IF (CH <> #0) THEN GOTO 53
 ELSE
 BEGIN
  CH := READKEY;
  IF ((CH = CHR(59)) OR (CH = CHR(77))) THEN
  BEGIN { F1 OR --> KEY }
   IF (TEMP[L] = CHR(13)) THEN BEGIN L := L - 1; GOTO 101; END;
   WRITE(TEMP[L]); GOTO 101;
  END;
  IF ((CH = CHR(60)) OR (CH = CHR(75)) OR (CH = CHR(83))) THEN
  BEGIN { F2 OR <-- OR Del KEY }
   IF (TEMP[L] = CHR(13)) THEN BEGIN L := L - 1; GOTO 101; END;
   FOR N := L + 1 TO 134 DO TEMP[N-1] := TEMP[N];
   L := L - 1; GOTO 101;
  END;
  IF ((CH = CHR(61)) OR (CH = CHR(79))) THEN
  BEGIN { F3 OR End KEY }
   IF (TEMP[L] = CHR(13)) THEN BEGIN L := L - 1; GOTO 101; END;
   REPEAT
    IF (TEMP[L] <> CHR(0)) THEN WRITE(TEMP[L]);
    L := L + 1;
   UNTIL ((TEMP[L] = CHR(13)) OR (L = 133));
   IF (L = 133) THEN BEGIN TEMP[133] := CHR(13); TEMP[134] := CHR(10); END;
   L := L - 1; GOTO 101;
  END;
 END;
53:FOR N := 133 DOWNTO L DO TEMP[N+1] := TEMP[N];
 TEMP[L] := CH; WRITE(TEMP[L]); GOTO 101;
81:FOR I := 1 TO L DO TEMP1[I] := TEMP[I];
 TEMP1[L] := CHR(13); TEMP1[L+1] := CHR(10); B := TEMP1; NEW_LINE := B; EXIT;
82:FOR I := 1 TO L DO TEMP1[I] := TEMP[I];
 TEMP1[L] := CHR(13); TEMP1[L+1] := CHR(10); B := TEMP1; NEW_LINE := B;
 FOR N := 133 DOWNTO 1 DO TEMP1[N+1] := TEMP1[N];
 TEMP1[1] := CH; EXIT;
83:IF (L <> 1) THEN
 BEGIN
  FOR I := 1 TO L DO TEMP1[I] := TEMP[I];
  TEMP1[L] := CHR(13); TEMP1[L+1] := CHR(10); B := TEMP1; NEW_LINE := B;
 END;
END;
PROCEDURE INSERT_APPEND;
LABEL 10, 20;
BEGIN
 TEXTCOLOR(CYAN);
 WRITELN('You can have up to 132 characters per line. ',
         'End each line with Enter key.');
 WRITELN('New line begins automatically after 132 characters ',
         'are entered in a line.');
 WRITELN('Use Ctrl d/D to end appending/inserting. ',
         'You can have maximum 3455 lines.');
 APPEND_INSERT_UPDATE_MESSAGE;
 WRITELN('Save(W/F) your work at regular intervals ',
         'to avoid the accidental loss of data.');
 TEXTCOLOR(YELLOW); WRITELN; TEMP[1] := CHR(13); L := 0;
 FOR I := 2 TO 134 DO TEMP[I] := CHR(0);
```

```
10:WRITE(NUMBER:4, ':');
 IF (L = 1) THEN WRITE(TEMP[1]);
 FOR I := 1 TO 134 DO TEMP1[I] := CHR(0); INSERT_APPEND1;
 IF ((CH = CHR(4)) AND (L <> 1)) THEN
 BEGIN
  INSERT_LINE(NUMBER, NEW_LINE); IF (NOT(AVAILABLE_MEMORY)) THEN GOTO 20;
  HIGHEST_NO := HIGHEST_NO + 1; NUMBER := NUMBER + 1; WRITELN;
 END;
 IF (CH = CHR(4)) THEN EXIT; INSERT_LINE(NUMBER, NEW_LINE);
 IF (NOT(AVAILABLE_MEMORY)) THEN GOTO 20;
 TEMP := TEMP1; HIGHEST_NO := HIGHEST_NO + 1;
 NUMBER := NUMBER + 1; WRITELN; IF (L = 133) THEN L := 1 ELSE L := 0; GOTO 10;
20:END;
PROCEDURE USAGE;
BEGIN
 TEXTCOLOR(CYAN);
 WRITELN('An Interactive Line Text Editor for Windows using Pascal 7.0 ',
         'By Prof. K.J.Satao');
 WRITELN; TEXTCOLOR(YELLOW);
 WRITELN('Usage : WPEDITOR<Enter> from the prompt or');
 WRITELN('        Double Click on WPEDITOR Icon or');
 WRITELN('        Select WPEDITOR Icon and Hit Enter key');
 WRITELN('(Alt & Enter keys give full screen view, pressing again gives title
          bar)');
END;
PROCEDURE NEW_FILE;
BEGIN
 NEW(HEAD); HEAD^.NEXT := NIL; HIGHEST_NO := 1; TOT_CH := 0; SAVED := TRUE;
 WRITELN(CLEAR1); IF (OPTION <> 'N') THEN USAGE;
 GET_CHECK_FILE_NAME1; ASSIGN(F, FILE_NAME); {$I-} RESET(F); {$I+}
 IF (IORESULT <> 0) THEN
 BEGIN
  {$I-} REWRITE(F); {$I+} TEXTCOLOR(CYAN);
  WRITELN('File...', FILE_NAME, ' is not found...');
  WRITELN('New file with size = 0 is created');
  WRITELN; CLOSE(F); TEXTCOLOR(YELLOW); WRITE('Hit any key to continue ');
  SOUND1; CH := READKEY; WRITELN; DIS_FILE_NAMES; DIS_MEM_CAP; WRITELN;
  WRITELN('Total lines = ', HIGHEST_NO - 1, ', Total characters = ', TOT_CH);
  WRITELN; WRITE('Hit any key to continue ');
  SOUND1; CH := READKEY; WRITELN;
 END
 ELSE
 BEGIN
  CLOSE(F); READ_LINES_FROM_FILE(HIGHEST_NO); TEXTCOLOR(CYAN);
  IF (NOT(AVAILABLE_MEMORY)) THEN
  BEGIN WRITELN('File truncated...'); WRITELN; END;
  WRITELN('File...', FILE_NAME, ' is opened for update');
  WRITELN; TEXTCOLOR(YELLOW); SOUND1; WRITE('Hit any key to continue ');
  CH := READKEY; WRITELN; DIS_FILE_NAMES;
  DIS_MEM_CAP; TEXTCOLOR(CYAN); WRITELN;
  WRITELN('Total lines = ', HIGHEST_NO - 1,
          ', Total characters loaded = ', TOT_CH);
  TEXTCOLOR(YELLOW); WRITELN; WRITE('Hit any key to continue ');
  SOUND1; CH := READKEY; WRITELN;
 END;
END;
```

```
PROCEDURE USE_W_OR_F;
BEGIN
 WRITELN; WRITELN('Use W or F to save i.e. write in a file...');
 WRITELN; WRITE('Hit any key to continue ');
 SOUND1; CH := READKEY; WRITELN;
END;
BEGIN { BODY OF PROGRAM STARTS }
 DATE_TIME; HOURS1 := HOURS; MINUTES1 := MINUTES; SECONDS1 := SECONDS;
 IF (PARAMCOUNT > 1) THEN
 BEGIN WRITELN; USAGE; WRITELN('Retry...'); WRITELN; HALT; END;
 TEXTBACKGROUND(RED); SETCBREAK(FALSE); CHECKBREAK := FALSE;
 FOR I := 1 TO 50 DO CLEAR1[I] := CHR(ORD(10)); NEW_FILE;
 REPEAT
 BEGIN
1:WRITELN; DATE_TIME; WRITE('File:'); TEXTCOLOR(CYAN); WRITE(FILE_NAME);
  WRITE(' '); TEXTCOLOR(YELLOW); WRITE('Date:'); TEXTCOLOR(CYAN);
  WRITE(DAY, '/', MONTH, '/', YEAR); WRITE(' '); TEXTCOLOR(YELLOW);
  WRITE('Day:'); TEXTCOLOR(CYAN); WRITE(DAY_NAM); WRITE(' '); TEXTCOLOR(YELLOW);
  WRITE('Time:'); TEXTCOLOR(CYAN); WRITELN(HOURS, ':', MINUTES, ':', SECONDS);
  TEXTCOLOR(YELLOW);
  WRITE('-------------------------------------------------------------------',
        '-----------------');
  TEXTCOLOR(CYAN);
  WRITE('Welcome to An Interactive Line Text Editor by Prof. K. J. Satao',
        ',Bhilai(CG),India');
  TEXTCOLOR(YELLOW);
  WRITE('-------------------------------------------------------------------',
        '-----------------');
  WRITE('Please choose an option...');
  TEXTCOLOR(CYAN); WRITE('C'); TEXTCOLOR(YELLOW); WRITE('-Clear screen/');
  TEXTCOLOR(CYAN); WRITE('A'); TEXTCOLOR(YELLOW); WRITE('-Append/');
  TEXTCOLOR(CYAN); WRITE('L'); TEXTCOLOR(YELLOW); WRITE('-List specific lines/');
  TEXTCOLOR(CYAN); WRITE('E'); TEXTCOLOR(YELLOW); WRITE('-Entirelist/');
  TEXTCOLOR(CYAN); WRITE('I'); TEXTCOLOR(YELLOW); WRITE('-Insert/');
  TEXTCOLOR(CYAN); WRITE('D'); TEXTCOLOR(YELLOW); WRITE('-Delete/');
  TEXTCOLOR(CYAN); WRITE('U'); TEXTCOLOR(YELLOW); WRITE('-Update/');
  TEXTCOLOR(CYAN); WRITE('W'); TEXTCOLOR(YELLOW); WRITE('-Write in open file/');
  TEXTCOLOR(CYAN); WRITE('F'); TEXTCOLOR(YELLOW); WRITE('-Write in any file/');
  TEXTCOLOR(CYAN); WRITE('O'); TEXTCOLOR(YELLOW); WRITE('-Copy/');
  TEXTCOLOR(CYAN); WRITE('M'); TEXTCOLOR(YELLOW); WRITE('-Move/');
  TEXTCOLOR(CYAN); WRITE('G'); TEXTCOLOR(YELLOW); WRITE('-Pagewise list/');
  TEXTCOLOR(CYAN); WRITE('V'); TEXTCOLOR(YELLOW); WRITE('-Convert case/');
  TEXTCOLOR(CYAN); WRITE('T'); TEXTCOLOR(YELLOW); WRITE('-Copy from other file/');
  TEXTCOLOR(CYAN); WRITE('S'); TEXTCOLOR(YELLOW); WRITE('-Search/');
  TEXTCOLOR(CYAN); WRITE('R'); TEXTCOLOR(YELLOW); WRITE('-Replace/');
  TEXTCOLOR(CYAN); WRITE('P'); TEXTCOLOR(YELLOW); WRITE('-Print a file/');
  TEXTCOLOR(CYAN); WRITE('Y'); TEXTCOLOR(YELLOW); WRITE('-Directory/');
  TEXTCOLOR(CYAN); WRITE('K'); TEXTCOLOR(YELLOW); WRITE('-Delete files/');
  TEXTCOLOR(CYAN); WRITE('H'); TEXTCOLOR(YELLOW); WRITE('-Change attributes/');
  TEXTCOLOR(CYAN); WRITE('J'); TEXTCOLOR(YELLOW); WRITE('-Search in files/');
  TEXTCOLOR(CYAN); WRITE('N'); TEXTCOLOR(YELLOW); WRITE('-Open new file/');
  TEXTCOLOR(CYAN + BLINK); WRITE('Q'); TEXTCOLOR(YELLOW);
  WRITE('-Quit(Upper/Lower Case) ');
  SOUND1; OPTION := UPCASE(READKEY); WRITELN(OPTION); WRITELN;
```

```
  IF ((OPTION <= '?') OR (OPTION >= 'Z') OR (OPTION = '@') OR
      (OPTION  = 'B') OR (OPTION = 'X')  OR (OPTION = CHR(0))) THEN
  BEGIN
   TEXTCOLOR(CYAN); WRITELN('Retry...'); WRITELN; TEXTCOLOR(YELLOW); GOTO 1;
  END;
  CASE OPTION OF
  'C':BEGIN
       WRITELN(CLEAR1);
      END;
  'E':BEGIN
       WRITELN('List entire file...E/e');
       WRITE('Hit any key to continue '); SOUND1;
       CH := READKEY; WRITELN; WRITELN;
       NUMBER := 1; STOP := HIGHEST_NO - 1;
       LIST_LINES(NUMBER, STOP); WRITELN;
       WRITELN('Listing over...Total characters including CR,LF = ', TOT_CH);
       WRITE('Hit any key to continue ');
       SOUND1; CH := READKEY; WRITELN;
      END;
  'I':BEGIN
       WRITELN('Insert lines...I/i'); WRITELN;
       WRITE('Before which line number ? '); SOUND1; READLN(A);
       VAL(A, NUMBER, CODE); NUMBER_CHECK(NUMBER);
       IF (NUMBER < 1) THEN NUMBER := 1;
       IF (NUMBER > HIGHEST_NO) THEN
       BEGIN
        WRITELN; INSERT_REQUIRED_BLANK_LINES(NUMBER);
        IF (NOT(AVAILABLE_MEMORY)) THEN GOTO 5;
       END;
       WRITELN; INSERT_APPEND;
5:     WRITELN; WRITELN;
       WRITE('Total lines = ', HIGHEST_NO - 1, ', Insertion over...',
             'Hit any key to continue ');
       SOUND1; CH := READKEY; WRITELN;
       DIS_MEM_CAP; SAVED := FALSE;
      END;
  'A':BEGIN
       WRITELN('Append Lines...A/a'); WRITELN;
       WRITELN('Starting appending...');
       WRITELN; NUMBER := HIGHEST_NO; INSERT_APPEND;
       WRITELN; WRITELN;
       WRITE('Appending over...Hit any key to continue ');
       SOUND1; CH := READKEY; WRITELN;
       DIS_MEM_CAP; SAVED := FALSE;
      END;
  'D':BEGIN
       WRITELN('Delete lines...D/d');
       IF (HIGHEST_NO > 1) THEN
       BEGIN
        WRITELN;
        WRITELN('Please note...this option will delete the desired lines.');
        WRITE('The deleted lines cannot be recovered later on unless the ');
        WRITELN('lines are written in afile before proceeding.');
        WRITELN; WRITE('Do you surely want to delete line(s) ? (Y/y for yes) ');
        SOUND1; CH := UPCASE(READKEY); WRITELN(CH);
```

```
      IF (CH = 'Y') THEN
      BEGIN
       GET_LINES;
       FOR I := NUMBER TO STOP DO DELETE_LINE(I);
       HIGHEST_NO := HIGHEST_NO - (STOP - NUMBER + 1);
       PREV := HEAD; LINK := PREV^.NEXT;
       NU1 := (STOP - NUMBER + 1);
       WHILE (LINK <> NIL) DO
       BEGIN
        NU := LINK^.LINE_NO;
        IF (NU > STOP) THEN NU := NU - NU1;
        LINK^.LINE_NO := NU; LINK := LINK^.NEXT;
       END;
       WRITELN;
       WRITE('Total lines remaining = ', HIGHEST_NO - 1, ', Deletion over...',
             'Hit any key to continue ');
       SOUND1; CH := READKEY; WRITELN;
       DIS_MEM_CAP; SAVED := FALSE;
      END
      ELSE USE_W_OR_F;
     END
     ELSE ZERO_LINES;
    END;
'L':BEGIN
     WRITELN('List from a line number to a line number...L/l');
     IF (HIGHEST_NO > 1) THEN
     BEGIN
      GET_LINES;
      WRITELN; LIST_LINES(NUMBER, STOP); WRITELN;
      WRITELN('Listing over...Total characters including CR,LF = ', TOT_CH);
      WRITE('Hit any key to continue ');
      SOUND1; CH := READKEY; WRITELN;
     END
     ELSE ZERO_LINES;
    END;
'O':BEGIN
     WRITELN('Copy & Paste lines...O/o');
     IF (HIGHEST_NO > 1) THEN
     BEGIN
      GET_LINES;
      WRITELN; WRITE('Paste before which line number ? '); SOUND1;
      READLN(A); VAL(A, SNO, CODE); NUMBER_CHECK(SNO);
      IF (SNO < 1) THEN SNO := 1;
      IF (SNO > HIGHEST_NO) THEN { INSERT REQUIRED BLANK LINES }
      BEGIN
       INSERT_REQUIRED_BLANK_LINES(SNO);
       IF (NOT(AVAILABLE_MEMORY)) THEN GOTO 10;
      END;
      COPY_LINES(NUMBER, STOP); L1 := SNO; LINK1 := HEAD1^.NEXT1;
      WHILE (LINK1 <> NIL) DO
      BEGIN { INSERT LINES IN MAIN TEXT FROM SECOND LINKED LIST }
       TEMP := LINK1^.LINE_TEXT1; NEW_LINE1 := TEMP;
       INSERT_LINE(L1, NEW_LINE1);
       IF (NOT(AVAILABLE_MEMORY)) THEN GOTO 10;
       HIGHEST_NO := HIGHEST_NO + 1; L1 := L1 + 1; LINK1 := LINK1^.NEXT1;
      END;
```

```
10:     WRITELN;
        WRITE('Total lines = ', HIGHEST_NO - 1, ', Copying & Pasting over...',
              'Hit any key to continue ');
        SOUND1; CH := READKEY; WRITELN;
        DIS_MEM_CAP; NEW(HEAD1); HEAD1^.NEXT1 := NIL;
        DISPOSE(HEAD1); { FREE SECOND LINKED LIST } SAVED := FALSE;
       END
       ELSE ZERO_LINES;
      END;
  'M':BEGIN
       WRITELN('Move lines(Cut & Paste)...M/m');
       IF (HIGHEST_NO > 1) THEN
       BEGIN
        GET_LINES;
        WRITELN; WRITE('Paste before which line number ? '); SOUND1;
        READLN(A); VAL(A, SNO, CODE); NUMBER_CHECK(SNO);
        IF (SNO < 1) THEN SNO := 1;
        IF (SNO > HIGHEST_NO) THEN
        BEGIN
         INSERT_REQUIRED_BLANK_LINES(SNO);
         IF (NOT(AVAILABLE_MEMORY)) THEN GOTO 15;
        END;
        IF (NOT((SNO >= NUMBER) AND (SNO <= STOP))) THEN
        BEGIN
         COPY_LINES(NUMBER, STOP); LINK1 := HEAD1^.NEXT1;
         FOR L1 := SNO TO SNO + STOP - NUMBER DO
         BEGIN { INSERT LINES IN MAIN TEXT FROM SECOND LINKED LIST }
          TEMP := LINK1^.LINE_TEXT1; NEW_LINE1 := TEMP;
          INSERT_LINE(L1, NEW_LINE1);
          IF (NOT(AVAILABLE_MEMORY)) THEN GOTO 15;
          HIGHEST_NO := HIGHEST_NO + 1; LINK1 := LINK1^.NEXT1;
         END;
         WRITELN;
         IF (SNO < NUMBER) THEN
         BEGIN
          START := NUMBER + (STOP - NUMBER + 1);
          STOP := STOP + (STOP - NUMBER + 1);
         END
         ELSE START := NUMBER;
         FOR I := START TO STOP DO DELETE_LINE(I); { DELETE COPIED LINES }
         HIGHEST_NO := HIGHEST_NO - (STOP - START + 1); PREV := HEAD;
         LINK := PREV^.NEXT; NU1 := STOP - START + 1;
         WHILE (LINK <> NIL) DO
         BEGIN
          NU := LINK^.LINE_NO; IF (NU > STOP) THEN NU := NU - NU1;
          LINK^.LINE_NO := NU; LINK := LINK^.NEXT;
         END;
15:      WRITE('Cut & Paste over...Hit any key to continue ');
         SOUND1; CH := READKEY;
         WRITELN; NEW(HEAD1); HEAD1^.NEXT1 := NIL;
         DISPOSE(HEAD1); { FREE SECOND LINKED LIST }
         SAVED := FALSE;
        END;
       END
       ELSE ZERO_LINES;
      END;
```

```
  'U':BEGIN
       WRITELN('Update a line...U/u'); WRITELN;
       WRITE('Which line number ? ');
       SOUND1; READLN(A); VAL(A, NUMBER, CODE);
       NUMBER_CHECK(NUMBER);
       WRITELN; IF (NUMBER < 1) THEN NUMBER := 1;
       IF (NOT(NUMBER > HIGHEST_NO - 1)) THEN
       BEGIN
        TEXTCOLOR(CYAN);
        WRITELN('You can have up to 132 characters in the line. ',
                'End the line with Enter key.');
        APPEND_INSERT_UPDATE_MESSAGE;
        WRITELN('Updation automatically stops after 132 characters ',
                'are entered in the line.');
        WRITELN; TEXTCOLOR(YELLOW); LIST_LINES(NUMBER, NUMBER);
        FOR I := 1 TO 134 DO TEMP1[I] := CHR(0);
        WRITE(NUMBER:4, ':'); L := 0;
        INSERT_APPEND1; DELETE_LINE(NUMBER); INSERT_LINE(NUMBER, NEW_LINE);
        IF (NOT(AVAILABLE_MEMORY)) THEN GOTO 20;
        WRITELN; WRITELN;
        WRITE('Updation over...Hit any key to continue ');
        SOUND1; CH := READKEY;
       END
       ELSE
       BEGIN
        TEXTCOLOR(CYAN);
        WRITELN('Invalid number...Maximum line number existing is...',
                 HIGHEST_NO - 1);
        WRITE('Updation aborted...Hit any key to continue ');
        SOUND1; CH := READKEY;
        TEXTCOLOR(YELLOW);
       END;
       WRITELN; DIS_MEM_CAP; SAVED := FALSE;
20:   END;
  'P':BEGIN
       WRITELN('Printing a file on the printer...P/p'); WRITELN;
       PRINT_FILE;
      END;
  'N':BEGIN
       WRITELN('Opening a new file...N/n'); WRITELN;
       WRITELN('Please note...this option will delete all current lines.');
       WRITELN('The deleted lines cannot be recovered later on.');
       WRITELN('You should proceed only after writing the current lines in a
                file.');
       WRITELN; WRITE('Do you surely want to proceed ? (Y/y for yes) ');
       SOUND1; READLN(CH); CH := UPCASE(CH);
       IF (CH = 'Y') THEN NEW_FILE ELSE USE_W_OR_F;
      END;
  'G':BEGIN
       WRITELN('Page wise list...G/g'); WRITELN;
       IF (HIGHEST_NO > 1) THEN
       BEGIN
        WRITE('Howmany lines per page ? (Maximum 22) '); SOUND1; READLN(A);
        VAL(A, NUMBER, CODE); NUMBER_CHECK(NUMBER);
        IF (NUMBER > 22) THEN NUMBER := 22;
        IF (NUMBER < 1) THEN NUMBER := 1;
```

```
        I := 1; STOP := HIGHEST_NO - 1;
        IF (NUMBER = 1) THEN
        BEGIN
         WRITELN;
         WRITE('Hit any key to continue ');
         SOUND1; CH := READKEY; WRITELN;
        END;
        REPEAT
         FVAL := I + NUMBER - 1; WRITELN(CLEAR1);
         IF (FVAL >= STOP) THEN FVAL := STOP;
         LIST_LINES(I, FVAL); I := FVAL + 1;
         TEXTCOLOR(CYAN); WRITELN;
         WRITE('Enter character S/s to stop or any other character to continue ');
         TEXTCOLOR(YELLOW); SOUND1; CH := UPCASE(READKEY);
        UNTIL ((FVAL = STOP) OR (CH = 'S'));
        WRITELN; WRITELN;
        WRITE('Display over...Hit any key to continue ');
        SOUND1; CH := READKEY; WRITELN;
      END
      ELSE ZERO_LINES;
     END;
'F':BEGIN
      WRITELN('Write specific lines in a specific file...F/f');
      IF (HIGHEST_NO > 1) THEN
      BEGIN
       GET_LINES;
       GET_CHECK_FILE_NAME1;
       WRITE_LINES_IN_FILE; DIS_FILE_NAMES;
       WRITELN;
      END
      ELSE ZERO_LINES;
     END;
'S':BEGIN
      WRITELN('Search a string...S/s');
      IF (HIGHEST_NO > 1) THEN
      BEGIN
       WRITELN; SEARCH_OPTIONS;
       IF (NOT(L = 0)) THEN
       BEGIN
        NUMBER := 1; STOP := HIGHEST_NO - 1;
        SEARCH_STRING(NUMBER, STOP, B12);
       END
       ELSE
       BEGIN
        TEXTCOLOR(CYAN); WRITELN;
        WRITELN('String length = 0...So no searching');
        TEXTCOLOR(YELLOW);
        WRITE('Hit any key to continue ');
        SOUND1; CH := READKEY;
        WRITELN;
       END;
      END
      ELSE ZERO_LINES;
     END;
```

```
'R':BEGIN
     WRITELN('Replace a string...R/r');
     IF (HIGHEST_NO > 1) THEN
     BEGIN
      WRITELN; SEARCH_OPTIONS;
      IF (NOT(L = 0)) THEN
      BEGIN
       M := 0;
       FOR I := 1 TO 20 DO TEMP12[I] := CHR(0);
       WRITELN('By which string ? (First 20 characters are considered)');
       SOUND1;
       WHILE (NOT EOLN) DO BEGIN M := M + 1; READ(REP_STRING[M]); END;
       READLN;
       IF (M > 20) THEN
       BEGIN
        TEXTCOLOR(CYAN);
        WRITELN('Number of characters > 20, ',
                'First 20 characters are considered');
        TEXTCOLOR(YELLOW); M := 20;
       END;
       FOR I := 1 TO M DO TEMP12[I] := REP_STRING[I];
       B13 := TEMP12; WRITELN; WRITELN;
       WRITE('Replace by query ? (Y/y for yes) '); SOUND1;
       QUERY := UPCASE(READKEY); WRITELN(QUERY);
       NUMBER := 1; STOP := HIGHEST_NO - 1;
       WRITELN; REPLACE_STRING(NUMBER, STOP, B12, B13);
      END
      ELSE
      BEGIN
       TEXTCOLOR(CYAN); WRITELN;
       WRITELN('String length = 0...So no replacement');
       TEXTCOLOR(YELLOW);
       WRITE('Hit any key to continue ');
       SOUND1; CH := READKEY; WRITELN;
      END;
      DIS_MEM_CAP; SAVED := FALSE;
     END
     ELSE ZERO_LINES;
    END;
'T':BEGIN
     WRITELN('Copy & Paste lines from another file...T/t');
     OLD_FILE_NAME := FILE_NAME;
     GET_CHECK_FILE_NAME1;
     FINDFIRST(FILE_NAME, FAANYFILE, DIRINFO);
     IF (DOSERROR = 0) THEN
     BEGIN
      WRITE('Paste before which line number ? ');
      SOUND1; READLN(A);
      VAL(A, NUMBER, CODE);
      NUMBER_CHECK(NUMBER);
      IF (NUMBER < 1) THEN NUMBER := 1;
      IF (NUMBER > HIGHEST_NO) THEN { INSERT REQUIRED BLANK LINES }
      BEGIN
       INSERT_REQUIRED_BLANK_LINES(NUMBER);
       IF (NOT(AVAILABLE_MEMORY)) THEN GOTO 25;
      END;
```

```
        READ_LINES_FROM_FILE(NUMBER); TEXTCOLOR(CYAN);
        IF (NOT(AVAILABLE_MEMORY)) THEN
        BEGIN
         WRITELN; WRITELN('File truncated...'); WRITELN;
        END;
        WRITELN('Total lines = ', HIGHEST_NO - 1,
                ', Copying & Pasting from another file over...');
       END
       ELSE
       IF (I = 18) THEN
       WRITELN('File does not exist/Wrong path...So no copying');
25:    TEXTCOLOR(YELLOW); FILE_NAME := OLD_FILE_NAME;
       WRITELN; SAVED := FALSE;
       WRITE('Hit any key to continue ');
       SOUND1; CH:=READKEY; WRITELN;
       DIS_MEM_CAP;
      END;
  'W':BEGIN
       WRITELN('Write entire file...W/w'); WRITELN;
       IF (HIGHEST_NO > 1) THEN
       BEGIN
        WRITELN('Writing in open file...', FILE_NAME);
        STOP := HIGHEST_NO - 1;
        NUMBER := 1; WRITE_LINES_IN_FILE;
        DIS_FILE_NAMES; SAVED := TRUE;
       END
       ELSE ZERO_LINES;
      END;
  'Y':BEGIN
       WRITELN('Directory...Y/y');
       OLD_FILE_NAME := FILE_NAME;
       DISPLAY_DIRECTORY;
       FILE_NAME := OLD_FILE_NAME;
      END;
  'J':BEGIN
       OLD_FILE_NAME := FILE_NAME;
       WRITELN('Search archive file(s) in a directory for a string ...J/j');
       SEARCH_FILES;
       FILE_NAME := OLD_FILE_NAME;
      END;
  'H':BEGIN
       WRITELN('Set file attribute(s)...H/h');
       SET_FILE_ATTRIBUTES;
       DIS_OPEN_FILE;
      END;
  'V':BEGIN
       WRITELN('Case convert...V/v');
       IF (HIGHEST_NO > 1) THEN
       BEGIN
        GET_LINES;
        TEXTCOLOR(CYAN); WRITELN;
        WRITE('Please enter...L/l for lower case or U/u for upper case or ',
              'T/t for title case orS/s for sentence case or any other character ',
              'to abort '); SOUND1; CHOICE := UPCASE(READKEY); WRITE(CHOICE);
        TEXTCOLOR(YELLOW);
```

```
      IF (NOT((CHOICE <> 'L') AND (CHOICE <> 'U') AND
              (CHOICE <> 'T') AND (CHOICE <> 'S'))) THEN
      BEGIN
       WRITELN;
       IF (CHOICE = 'L') THEN LOWER_CASE(NUMBER, STOP)
       ELSE IF (CHOICE = 'U') THEN UPPER_CASE(NUMBER, STOP)
       ELSE IF ((CHOICE = 'T') OR (CHOICE = 'S')) THEN
       BEGIN
        LOWER_CASE(NUMBER, STOP);
        TITLE_SENTENCE_CASE(NUMBER, STOP);
       END;
       WRITELN;
       WRITE('Case is converted...Hit any key to check ');
       SOUND1; CH := READKEY;
       SAVED := FALSE; WRITELN; WRITELN;
       LIST_LINES(NUMBER, STOP);
      END
      ELSE
      BEGIN
       WRITELN; WRITELN('Choice is not L/l or U/u or T/t or S/s...');
       WRITE('Hence...Case is not changed...Hit any key to continue ');
       SOUND1; CH := READKEY; WRITELN;
      END;
     END
     ELSE ZERO_LINES;
    END;
'K':BEGIN
     WRITELN('Delete one or more archive files...K/k');
     OLD_FILE_NAME := FILE_NAME;
     DELETE_FILES; WRITELN;
     WRITE('Hit any key to check '); SOUND1; CH := READKEY;
     WRITELN; DISPLAY_DIRECTORY;
     FILE_NAME := OLD_FILE_NAME;
     DIS_OPEN_FILE;
    END;
'Q':BEGIN
     WRITELN('Quit...Q/q'); WRITELN;
     IF (SAVED) THEN
     BEGIN
      NEW(HEAD); DISPOSE(HEAD); { FREE FIRST LINKED LIST } SAVE := 'N';
     END
     ELSE
     BEGIN
      TEXTCOLOR(CYAN);
      WRITELN('Text is possibly changed but not saved');
      WRITELN;
      WRITE('Do you want to save it(write in a file) ? (Y/y for yes) ');
      TEXTCOLOR(YELLOW); SOUND1;
      SAVE := UPCASE(READKEY); WRITELN(SAVE);
      IF (SAVE = 'Y') THEN USE_W_OR_F
      ELSE
      BEGIN
       DISPOSE(HEAD); { FREE FIRST LINKED LIST } WRITELN;
      END;
     END;
```

```
       IF (SAVE <> 'Y') THEN
       BEGIN
        TEXTCOLOR(CYAN); DATE_TIME;
        HOURS2 := HOURS; MINUTES2 := MINUTES; SECONDS2 := SECONDS;
        WRITELN('Starting Time(HH:MM:SS) : ',
                 HOURS1:2,':',MINUTES1:2,':',SECONDS1:2);
        WRITELN('Ending   Time(HH:MM:SS) : ',
                 HOURS2:2,':',MINUTES2:2,':',SECONDS2:2);
        WRITELN;
        IF (SECONDS2 < SECONDS1) THEN
        BEGIN
         SECONDS11 := SECONDS2 + 60 - SECONDS1;
         MINUTES1 := MINUTES1 + 1;
        END
        ELSE SECONDS11 := SECONDS2 - SECONDS1;
        IF (MINUTES2 < MINUTES1) THEN
        BEGIN
         MINUTES11 := MINUTES2 + 60 - MINUTES1;
         HOURS1 := HOURS1 + 1;
        END
        ELSE MINUTES11 := MINUTES2 - MINUTES1;
        HOURS11 := HOURS2 - HOURS1;
        IF (HOURS11 < 0) THEN HOURS11 := HOURS11 + 24;
        WRITELN('Elapsed  Time(HH:MM:SS) : ',
                 HOURS11:2, ':', MINUTES11:2 , ':', SECONDS11:2);
        WRITELN; TEXTCOLOR(YELLOW);
        WRITE('See you again...Bye...Hit any key ');
        CH := READKEY; TEXTCOLOR(WHITE);
        TEXTBACKGROUND(BLACK); WRITELN; WRITELN;
        HALT;
       END;
      END; { END OF CASE Q/q }
     END;  { END OF CASE }
    END;   { END OF REPEAT }
 UNTIL ((OPTION = 'Q') AND (SAVE <> 'Y'));
END. { BODY OF PROGRAM ENDS }
```