



## Design & Development of a New and Efficient Approach for Line Text Editing in Turbo Pascal 5.5 for Windows/Netware/MS-DOS\*

K. J. Satao

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

[kjsatao@rediffmail.com](mailto:kjsatao@rediffmail.com), [kjsatao@gmail.com](mailto:kjsatao@gmail.com), [kjsatao@indiatimes.com](mailto:kjsatao@indiatimes.com)

**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 and EDIT also work in Windows. MS-DOS had very good word processors viz. Word Star, Word Perfect, etc. All the editors which work in MS-DOS, also work in the Novell's Netware OS as well. 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 5.5 for Windows/Netware/MS-DOS 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 which works like an IDE, Integrated Development Environment, available in most of the programming languages viz. Turbo C, Turbo Pascal, etc. 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 and shall work efficiently on Novell's Netware and MS-DOS as well. This research work has, in all, twenty eight 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.

### I. 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, “B” for taking backup, “X” for moving files from one directory to the other, “Z” for renaming files, “@” for locating files in a drive, “?” for going to DOS shell, “N” for opening a new file, and “Q” for quitting. Please note that all the English alphabets are used and help had to be taken from the non alphabetic characters. With the help of the DOS shell, a user can compile his program and return to the editor for further processing or he may execute DOS commands if required.

Apart from many good features, the four most important features of this research work are...

- This work is compatible with text processing utilities such as EDLIN, EDIT, Notepad, etc.
- Insertion/appending/updation is made very easy. The program takes care of RO, H, S, D, A, Vol Id files.
- 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.
- The program works like an IDE and a user is can manipulate many file manipulation tasks.

\*Orally presented and demonstrated in the National Conference on “Information Technology, Communication and Development”, organized by Rungta College of Engineering & Technology, Bhilai, C.G., India, held on 30<sup>th</sup> March 2007.

### II. IMPORTANT EDITING PROCEDURES AND MESSAGES INCORPORATED

*[Modified, Windows/Netware/MS-DOS version of Satao, 1992 and 1994]*

#### a. Procedure USAGE

'Interactive Line Text Editor for Windows/Netware/MS-DOS in Pascal 5, By Satao'

'Usage : NPEDITOR<Enter> from the prompt or'

' Double Click on NPEDITOR Icon or'

' Select NPEDITOR Icon and Hit Enter key'

'(Alt & Enter keys give full screen view, pressing again gives title bar)'

#### b. Procedure GET\_DRIVE\_NAME

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

**c. 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. PASCAL5\NPEDITOR.PAS'  
 '78 characters maximum(case insensitive e.g. TeST.c = TEST.C)'

**d. Procedure DIS\_MEM\_CAP; { DISPLAY MEMORY CAPACITY}**

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

**e. 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).'

**f. 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 / B - Backup/ X-Move files/Z-Rename files/@-Locate files / ?-DOS shell/ N-Open new file/Q-Quit(Upper/Lower Case)'

**g. 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'

**h. 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.'

**i. 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 1357 lines.'  
 APPEND\_INSERT\_UPDATE\_MESSAGE  
 'Save(W/F) your work at regular intervals to avoid the accidental loss of data.'

**j. 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.'

**k. Procedure GET\_LINES**

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

**l. 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)'

**m. 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'

**n. 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)'

**o. Replace Message**

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

**p. 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)

**q. Procedure NEW\_FILE**

'Please note...this 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)'

**III. RESULTS & DISCUSSION**

In Computer Science, a Linked List is one of the fundamental Data Structures. It consists of a sequence of Nodes, each containing arbitrary Data Fields and one or two References ("Links") pointing to the Next and/or Previous Nodes. The principal benefit of a Linked List over a conventional Array is that the order of the Linked items may be different from the order that the data items are stored in memory, allowing the list of items to be traversed in a different order.

A linked list is a self-referential data type because it contains a pointer or link to another datum of the same type. Linked lists permit insertion and removal of nodes at any point in the list, but do not allow random access. Several different types of linked list exist: singly-linked lists, doubly-linked lists, and circularly-linked lists. Linked lists can be implemented in most languages. Languages such as Lisp have the data structure built in, along with operations to access the linked list. Procedural or object-oriented languages such as Pascal, C, C++, and Java also permit us to create linked lists and operate on them.

In this paper I have used singly-linked lists to develop a new, interactive, and more user friendly approach to line text editing for document / non document(program) files. 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 work is an innovative one. The program works like an IDE, Integrated Development Environment, available in some of the programming languages viz. Turbo C, Turbo Pascal, etc. but the new feature is interactivensess.

#### IV. 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.

#### V. CONCLUSION

Even today, some big companies are involved in the development of IDEs(Integrated Development Environments). Many IDEs are existing from many companies e.g. Microsoft Visual Interdev, JBuilder, Power Builder, Eclipse, JDeveloper, Microsoft Visual Studio, etc. The famous products of IBM are WebSphere Studio, Rational Application Developer, Rational Software Architect (RSA), etc. The IDEs are expected to work for C, C++, Java, J2EE, etc.

This paper relates to the design and development of an interactive IDE. I have tried to give my own approach for editing and have designed and developed my own IDE. The existing IDEs are language specific IDEs. But my IDE is general purpose IDE i.e. it does not restrict to any single language. This IDE shall be very useful to all the new comers in the Computer Science & Engineering discipline. In developing this IDE a total interactive approach is used. Thus the IDE shall prove to be a self learning and self guiding tool, without requiring any specific training.

#### VI. ACKNOWLEDGEMENT

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.

#### VII. 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 : SOURCE CODE OF THE PROGRAM

*[Modified, Windows/Netware/MS-DOS version of Satao, 1992 and 1994]*

```

{$M 8192,0,190000}{STACKSIZE,HEAPMIN,HEAPMAX}
{Final version}
PROGRAM NPEDITOR;
USES CRT,DOS;
LABEL 1,5,10,15,20,25;
CONST
  RESERVED_FILE_NAMES : ARRAY[1..13] OF STRING[6] =
    ('AUX','CLOCK$','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;
VAR
  IGNORE_CASE,QUERY,WILD,OPTION,CHOICE,CH,OLD_CH,
  OVER_FLOW,SAVE,DR_NAME : CHAR;
  FILE_NAME,OLD_FILE_NAME,ASCII_Z,F_NAME,F_NAME1,
  F_NAME2 : PATHSTR;
  NUMBER,START,STOP,HIGHEST_NO,HIGHEST_NO1,CODE,
  LINE_OCC,DR_NO : INTEGER;
  NEW_LINE,NEW_LINE1 : TEXT1; F,PRINTER : TEXT;
  D1,D2,D3 : DIRSTR; HEAD,LINK,PREV,P : POINTER1;
  TEMP,TEMP1,TEMP2,B : TEXT1; CLEAR1 : CLEAR;
  S1 : STRING; BUF : ARRAY[1..10240] OF CHAR;
  DIRINFO : SEARCHREC;

```

```

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;
N1, N2, N3 : NAMESTR; E1, E2, E3 : EXTSTR;
M, N, I, J, K, I1, J1, K1, NU, NU1, OCC, L, L1, FVAL, SNO,
LINE_REP : INTEGER; COMMAND : COMSTR;
DAY, MONTH, YEAR, DAYOFWEEK, HOURS, MINUTES, SECONDS,
HUNDREDTHS : WORD;
HOURS1, MINUTES1, SECONDS1, HOURS2, MINUTES2,
SECONDS2, ATTR : WORD; A : STRING[5];
HOURS11, MINUTES11, SECONDS11 : INTEGER;
DAY_NAM : STRING[3];
PAGE_HEADING_CHARS, PAGE_HEADING_LINES, T_CH,
LINE_OCC1, LINENUM, PGNO : INTEGER;
PROCEDURE SPLIT_FILE_NAME;
BEGIN
FOR I:=1 TO LENGTH(FILE_NAME) 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);
END; FSPLIT(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;
IF (N1='{NULL}') THEN
BEGIN
WRITELN; WRITELN('Invalid file name...Primary
file name cannot be null');
INVALID_FILE_NAME:=TRUE; EXIT;
END;
FOR I:=1 TO 13 DO
IF ((N1=RESERVED_FILE_NAMES[I]) AND (E1=''))
THEN
BEGIN{CHECKING WHETHER PRIMARY FILE NAME IS A
RESERVED WORD OR NOT}
WRITELN; WRITELN('Invalid file name...It cannot
be a reserved word');
INVALID_FILE_NAME:=TRUE; EXIT;
END;
FOR I:=1 TO LENGTH(N1) DO
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; EXIT;
END; {CHECKING WHETHER PRIMARY FILE NAME CONTAINS
THE PERMITTED CHARACTERS OR NOT}
IF (E1='{NULL}') THEN EXIT;
FOR I:=2 TO LENGTH(E1) DO{DOT IS A PART OF
EXTENSION, HENCE START WITH 2}
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; EXIT;
END; {CHECKING WHETHER SECONDARY FILE NAME
CONTAINS THE PERMITTED CHARACTERS OR NOT}
END;
PROCEDURE DISPLAY_FILE_ATTRIBUTES;
BEGIN
ASSIGN(F, F_NAME); GETFATTR(F, ATTR);
IF (DOSERROR<>0) THEN
WRITE('Error code = ', DOSERROR)
ELSE
BEGIN
IF (ATTR AND ARCHIVE<>0) THEN
WRITE('<Archive>');
IF (ATTR AND READONLY<>0) THEN
WRITE('<Read Only>');
IF (ATTR AND HIDDEN<>0) THEN
WRITE('<Hidden>');
IF (ATTR AND SYSDIR<>0) THEN
WRITE('<System>');
IF (ATTR AND DIRECTORY<>0) THEN
WRITE('<Directory>');
IF (ATTR AND VOLUMEID<>0) THEN
WRITE('<Volume Id>');
END;
WRITELN;
END;
PROCEDURE DIS_OPEN_FILE; {DISPLAY OPEN FILE}
BEGIN
F_NAME:=FILE_NAME; WRITELN;
FINDFIRST(FILE_NAME, ANYFILE, DIRINFO);
WRITELN('Open file name = ', FILE_NAME); WRITELN;
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. ');
CH:=READKEY; DR_NAME:=UPCASE(CH); WRITE(DR_NAME);
UNTIL (ORD(DR_NAME)>=65) AND (ORD(DR_NAME)<=90);
END;
PROCEDURE GET_PATH;
LABEL 10;
BEGIN
M:=LENGTH(S1)-1;
IF ((L=0) AND (S1[M]<>'\')) THEN
BEGIN ASCIIZ:=S1+'\*. *'; GOTO 10; END;
IF ((L=0) AND (S1[M]='\')) THEN
BEGIN ASCIIZ:=S1+'. *'; GOTO 10; END;
IF ((D1='') AND (S1[M]<>'\')) THEN
BEGIN ASCIIZ:=S1+'\'+ASCIIZ; GOTO 10; END;
IF ((D1='') AND (S1[M]='\')) THEN
BEGIN ASCIIZ:=S1+ASCIIZ; GOTO 10; END;
IF ((D1='\') AND (S1[M]<>'\')) THEN
BEGIN ASCIIZ:=DR_NAME+' '+ASCIIZ; GOTO 10; END;
IF ((D1='\') AND (S1[M]='\')) THEN
BEGIN ASCIIZ:=S1+N1+E1; GOTO 10; END;
IF (ASCIIZ[L-1]='\') THEN
BEGIN ASCIIZ:=ASCIIZ+'*. *'; GOTO 10; END;
ASCIIZ:=DR_NAME+':\'+ASCIIZ;
10:FILE_NAME:=ASCIIZ;
SPLIT_FILE_NAME; TEST_FILE_NAME;
ASCIIZ:=FILE_NAME;
END;
PROCEDURE CHECK_ERROR;
BEGIN
I:=DOSERROR; J:=0; 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;
IF ((I=3) OR (I=6) OR (I=8) OR (I=10) OR (I=11))
THEN
BEGIN
WRITELN;
WRITE('Retry...Hit any key to continue ');
CH:=READKEY; WRITELN; J:=1;
END;
END;

```

```

PROCEDURE GET_FILE_NAME_MESSAGE;
BEGIN
  DR_NO:=ORD(DR_NAME)-64; WRITELN;
  WRITELN('Please give file name...');
  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.
  PASCAL5\NPEDITOR.PAS');
  WRITE('78 characters maximum(Case insensitive
  e.g. TeSt.c=TEST.C) ');
END;
PROCEDURE GET_FILE_NAME1;
BEGIN
  GET_FILE_NAME_MESSAGE;
  READLN(ASCIIIZ); L:=LENGTH(ASCIIIZ);
  {FILENAME CANNOT HAVE \ / : * " < > | ? + etc.}
  IF (L=0) THEN
  BEGIN
    WRITELN;
    WRITELN('Invalid file name...File name cannot be
    null');
    INVALID_FILE_NAME:=TRUE; EXIT;
  END;
  FOR I:=1 TO L DO
  IF ((ASCIIIZ[I]='*') OR (ASCIIIZ[I]='?')) THEN
  BEGIN
    INVALID_FILE_NAME:=TRUE; WRITELN;
    WRITELN('Invalid file name...File name cannot
    have * or ?');
    EXIT;
  END;
  FSPLIT(ASCIIIZ,D1,N1,E1);
  GETDIR(DR_NO,S1); 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,ANYFILE,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 ');
  READLN(ASCIIIZ); L:=LENGTH(ASCIIIZ);{FILENAME
  CANNOT HAVE \ / : " < > | + etc.}
  FSPLIT(ASCIIIZ,D1,N1,E1); GETDIR(DR_NO,S1);
  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,ANYFILE,DIRINFO);
  CHECK_ERROR; IF (J=1) THEN GOTO 5;
END;
PROCEDURE GET_NO_OF_CHARS;
BEGIN
  I:=135; REPEAT I:=I-1;
  UNTIL (TEMP[I]<>CHR(0));
END;
PROCEDURE DIRECTORY_HEADING;
BEGIN
  IF (OPTION='Y') THEN WRITELN(CLEAR1) ELSE
  WRITELN; WRITELN('Directory : ',ASCIIIZ);
  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(ASCIIIZ,ANYFILE,DIRINFO);
  {ALL FILE TYPES...RO, H, S, D, A, AND VOL ID ARE
  LISTED}
  WHILE (DOSERROR=0) DO
  BEGIN
    COUNT:=COUNT+1;
    WRITE(COUNT:4,' ',DIRINFO.NAME:12,' ',
    DIRINFO.SIZE:10,' ');
    F_NAME:=D1+DIRINFO.NAME;
    DISPLAY_FILE_ATTRIBUTES;
    I:=COUNT DIV 20;
    IF (COUNT=I*20) THEN
    BEGIN
      WRITE('-----');
      WRITELN('-----',
      'Hit any key...');
      CH:=READKEY; DIRECTORY_HEADING;
      END; FINDNEXT(DIRINFO);
    END;
    WRITE('-----');
    WRITELN('-----',
    'Hit any key...');
    CH:=READKEY; WRITELN;
    WRITELN('Total number of files = ',COUNT,
    ', Use DIR in DOS Shell(?) to know Disk
    Free Space...');
  END;
PROCEDURE DIS_FILE_NAMES;{DISPLAY FILE NAMES}
BEGIN
  FSPLIT(FILE_NAME,D1,N1,E1);
  ASCIIIZ:=D1+N1+'*.*'; DISPLAY_DIRECTORY;
END;
PROCEDURE DIS_MEM_CAP;{DISPLAY MEMORY CAPACITY}
BEGIN
  WRITELN; WRITELN('Main memory available = ',
  MEMAVAIL,' Bytes');
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));
    WRITELN('Original file contents...',
    FILESIZE(FROM_FILE),
    ' Bytes...copied in...');
    WRITELN(F_NAME); CLOSE(FROM_FILE);
    CLOSE(TO_FILE); WRITELN;
    WRITE('Hit any key to continue ');
    CH:=READKEY; WRITELN;
  END;

```

```

PROCEDURE LESS_MEMORY;
BEGIN
  Writeln; Writeln('Insufficient memory...Could not
    execute COMMAND.COM');
  Writeln('Compile, Make .EXE, Run and/or Delete a
    few lines temporarily and/or ',
    'close other applications to release
    memory');
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,ANYFILE,DIRINFO);
  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 READONLY<>0) THEN
      BEGIN Write('<Read Only>'); C_W:=1; END;
      IF (ATTR AND HIDDEN <>0) THEN
      BEGIN Write('<Hidden>'); C_W:=1; END;
      IF (ATTR AND SYSFILE <>0) THEN
      BEGIN Write('<System>'); C_W:=1; END;
      IF (ATTR AND DIRECTORY<> 0) THEN
      BEGIN Write('<Directory>'); C_W:=1; END;
      IF (ATTR AND VOLUMEID<>0) THEN
      BEGIN Write('<Volume Id>'); C_W:=1; END;
      IF (C_W=1) THEN
      BEGIN
        Write(' file...Cannot write...Hit any key to
          continue ');
        CH:=READKEY; Writeln; EXIT;
      END;
    END;
  END; 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,ANYFILE,DIRINFO);
    Writeln('File...',FILE_NAME,
      ' , Size...',DIRINFO.SIZE,' Bytes');
    Write('Already exists...Overwrite it(old
      contents will be replaced) ? ',
      '(Y/y for yes) ');
    CH:=UPCASE(READKEY); Writeln(CH);
    IF (CH<>'Y') THEN
    BEGIN FILE_NOT_DISTURBED; EXIT; END;
    Writeln; FSPLIT(FILE_NAME,D1,N1,E1);
    F_NAME:=D1+N1+'.OLD'; COPY_OLD_FILE;
  END;
  DIS_FILE_NAMES; Writeln;
  Write('Do you surely want to write ? (Y/y for
    yes) ');
  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;
    Writeln('Total lines written = ',STOP-START+1,
      ' , Total characters written = ',TOT_CH);
    Writeln;
    Write('Writing over...Hit any key to check ');
    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
    Writeln; Writeln('Insufficient memory to insert
      line in main linked list...');
    Write('Hit any key to continue ');
    CH:=READKEY; 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
    Writeln; Writeln('Insufficient memory to insert
      line in second linked list...');
    Write('Hit any key to continue ');
    CH:=READKEY; 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
TEMP[NU]:=CH;
IF (MEMAVAIL<268) THEN
BEGIN
WRITELN; WRITELN('Insufficient memory to
load/copy file...Use other editor. ');
WRITE('Hit any key to continue ');
CH:=READKEY; AVAILABLE_MEMORY:=FALSE;
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;
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; WRITELN('Number of occurrences in
this line = ',LINE_OCC);
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 '); CH:=UPCASE(READKEY);
IF (CH='S') THEN BEGIN WRITELN; GOTO 1; END;
WRITELN(CLEAR1);
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 ');
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';
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
WRITELN('Line over flow...Last ',M-L,
' Character(s) will be lost...');
WRITE('Original line occurrence = ',LINE_OCC1,
', Replace ? (Y/y for yes...A/a to abort) ');
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) DOWNT0 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]:=CHR(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 DOWNT0 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('Total occurrences = ',TOT_OCC);
WRITELN;
WRITELN('Total replacements done = ',REP_DONE);
WRITELN;
WRITE('Replacement over...Hit any key to
continue '); 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;
    ASCIIZ:=FILE_NAME+CHR(0);
    ASSIGN(F,ASCIIZ); SETTEXTBUF(F,BUF);
    {$I-} RESET(F); {$I+}
    IF (IORESULT<>0) THEN
    BEGIN
        WRITELN('File does not exist/Wrong path so no
            printing'); GOTO 6;
    END; CLOSE(F);
    REPEAT
        WRITELN('Please save your work before
            proceeding...Runtime error may occur...');
        WRITE('Choose...1..To print or 2..To abort ');
        CH:=READKEY; WRITELN(CH); 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; 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; WRITELN('Invalid choice...Re-enter');
            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) ');
    CH1:=READKEY; WRITELN(CH1);
    IF (CH1=CHR(0)) THEN CH1:='N';
    IF (UPCASE(CH1)='Y') THEN
    BEGIN
        L:=LENGTH(FILE_NAME); 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;
                                        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 ');
                                        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 : REGISTERS; 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;
                                        FILE_NAME:=ASCIIZ+CHR(0);
                                        AH:=CHECK_ATTRIBUTES; DS:=SEG(FILE_NAME[1]);
                                        DX:=OFS(FILE_NAME[1]); AL:=GET_ATTRIBUTES;
                                        MSDOS(REG);
                                        IF ((FLAGS AND CARRY_FLAG)=1) THEN
                                            BEGIN
                                                WRITELN; WRITELN('Unable to find...',
                                                    FILE_NAME,'...Job aborted'); 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;
  WRITELN; WRITELN;
  WRITE('Do you want to change the attributes ?
    (Y/y for yes) ');CH:=READKEY; WRITELN(CH);
  IF (UPCASE(CH)='Y') THEN
    BEGIN
      WRITELN; REG.AH:=CHECK_ATTRIBUTES;
      REG.DS:=SEG(FILE_NAME[1]);
      REG.DX:=OFS(FILE_NAME[1]);
      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; WRITELN('Cannot change attribute...');
      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) ');
      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) ');
      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) ');
      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) ');
      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) ');
      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) ');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) ');
    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) ');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 ');
  CH:=READKEY; WRITELN; DISPLAY_DIRECTORY;
  FILE_NAME:=OLD_FILE_NAME;
END;
PROCEDURE SEARCH_FILES;
LABEL 10,12;
VAR FILEVAR : TEXT; STRING; FOUND : BOOLEAN;
    COUNT,COUNT1,COUNT2,LINENUMBER : INTEGER;
    ONELINE,SUBSTRING,ONELINE_OLD,SUBSTRING_OLD :
  BEGIN
    GET_CHECK_FILE_NAME2; COUNT:=0; COUNT1:=0;
    COUNT2:=0; WRITELN;
    WRITE('Search for(Maximum 255 characters) ? (Hit
      just Enter to quit) ');READLN(SUBSTRING);
    IF (LENGTH(SUBSTRING)=0) THEN
      BEGIN WRITELN; GOTO 10; END;
    SUBSTRING_OLD:=SUBSTRING; WRITELN;
    WRITE('Do you want to ignore case?(Y/y foryes)');
    IGNORE_CASE:=UPCASE(READKEY);
    WRITELN(IGNORE_CASE); WRITELN;
    IF (IGNORE_CASE='Y') THEN
      FOR I:=1 TO LENGTH(SUBSTRING) DO
        SUBSTRING[I]:=UPCASE(SUBSTRING[I]);
      WRITELN('Searching...',ASCIIZ,' for...',
        SUBSTRING_OLD); WRITELN;
      FSPLIT(ASCIIZ,D1,N1,E1);
      FINDFIRST(ASCIIZ,ARCHIVE,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');
          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
                      WRITELN; WRITE('Eight lines displayed...Hit
                        any key '); CH:=READKEY; WRITELN;
                    END;
                  END;
                END;
            END;
          END;
        END;

```

```

IF (FOUND) THEN
BEGIN
  WRITELN('Number of lines with the string in the
  file = ',COUNT1); WRITELN;
  WRITE('Enter S/s to stop or other character to
  continue '); 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 ');
  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
  FSPLIT(ASCIIIZ,D1,N1,E1); C_D:=0;
  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 READONLY<>0) THEN
    BEGIN WRITE('<Read Only>');C_D:=1; END;
    IF (ATTR AND HIDDEN<>0) THEN
    BEGIN WRITE('<Hidden>'); C_D:=1; END;
    IF (ATTR AND SYSFILE<>0) THEN
    BEGIN WRITE('<System>'); C_D:=1; END;
    IF (ATTR AND DIRECTORY<> 0) THEN
    BEGIN WRITE('<Directory>');C_D:=1; END;
    IF (ATTR AND VOLUMEID<>0) THEN
    BEGIN WRITE('<Volume Id>');C_D:=1; END;
    IF (C_D=1) THEN
    BEGIN
      WRITELN(' file...');
      WRITE('Cannot delete...Hit any key to
      continue'); CH:=READKEY; WRITELN; EXIT;
    END;
    IF (ATTR AND ARCHIVE<>0) THEN
    BEGIN ERASE(F);I:=I+1; WRITE('...Deleted'); END;
  END;
END;
BEGIN
  GET_CHECK_FILE_NAME2;
  FINDFIRST(ASCIIIZ,ANYFILE,DIRINFO);
  IF (DOSERROR<>0) THEN
  BEGIN
    WRITE('No such file(s) exist...');I:=0; GOTO 20;
  END;
  WRITELN('Confirm delete');
  WRITE(ASCIIIZ,' ? ', '(Y/y for yes) ');
  CH:=UPCASE(READKEY); WRITELN(CH); I:=0;
  IF (CH<>'Y') THEN
  BEGIN WRITELN;WRITELN('Deletion aborted...'); END
  ELSE
  BEGIN
    IF ((POS('*',ASCIIIZ)>0) OR (POS('?',ASCIIIZ)>0))
    THEN
    BEGIN
      WRITELN;WRITE('Delete by query?(Y/y for yes)');
      CH:=UPCASE(READKEY); WRITELN(CH);
      IF (CH='Y') THEN GOTO 10;
    END; FINDFIRST(ASCIIIZ,ANYFILE,DIRINFO);
    WHILE (DOSERROR=0) DO
    BEGIN
      WRITELN; WRITE('File - ',DIRINFO.NAME:12);
      ERASE_FILE; FINDNEXT(DIRINFO);
    END;
    GOTO 20;
  END;
  10:FINDFIRST(ASCIIIZ,ANYFILE,DIRINFO);
  WHILE (DOSERROR=0) DO
  BEGIN
    WRITELN; WRITE('Delete File - ',
    DIRINFO.NAME:12,' ? (Y/y for yes) ');
    CH:=UPCASE(READKEY); WRITE(CH);
    IF (CH='Y') THEN ERASE_FILE;
    FINDNEXT(DIRINFO);
  END;
  20:WRITELN; WRITELN;
  WRITELN('Number of files deleted = ',I);
  END;
END;
PROCEDURE NUMBER_CHECK(NUMBER1:INTEGER);
BEGIN
  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;
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 ? ');
  READLN(A); VAL(A,NUMBER,CODE);
  NUMBER_CHECK(NUMBER);
  IF (NUMBER<1) THEN NUMBER:=1;
  WRITE('To which line number ? ');
  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
    WRITELN; WRITELN('Invalid range of numbers...
    From > To or From > Highest number');
    WRITELN('Maximum line number existing = ',
    HIGHEST_NO-1,'. Retry...'); GOTO 5;
  END;
END;
PROCEDURE ZERO_LINES;
BEGIN
  WRITELN;
  WRITELN('Maximum line number existing = 0');
  WRITE('Hit any key to continue '); CH:=READKEY;
  WRITELN;
END;
PROCEDURE SEARCH_OPTIONS;
BEGIN
  WRITE('Do you want to ignore case?(Y/y for yes)');
  IGNORE_CASE:=UPCASE(READKEY);
  WRITELN(IGNORE_CASE); WRITELN;
  WRITE('? can be used as a wild character ',
  'e.g. s???o shall match with satao, spgro, etc.',
  'Be careful in using it...Use it?(Y/y for yes)');
  WILD:=UPCASE(READKEY); WRITELN(WILD); 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) ');

```

```

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
BEGIN{BACKSPACE}
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
BEGIN{HORIZONTAL TAB}
FOR N:=129 DOWNT0 L DO TEMP[N+5]:=TEMP[N];
FOR I:=L TO L+4 DO
FOR 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 DOWNT0 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 DOWNT0 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
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 1357 lines. ');
APPEND_INSERT_UPDATE_MESSAGE;
WRITELN('Save(W/F) your work at regular intervals
to avoid the accidental loss of data. ');
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
WRITELN('Interactive Line Text Editor for
Windows/Netware/MS-DOS using Pascal 5, '
' By Satao');
WRITELN('Usage:NPEDITOR<Enter> from the prompt
or');
WRITELN(' Double Click on NPEDITOR Icon
or');
WRITELN(' Select NPEDITOR 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+}
WRITELN('File...',FILE_NAME,' is not found...');
WRITELN('New file with size = 0 is created');
CLOSE(F);
WRITE('Hit any key to continue ');
CH:=READKEY;WRITELN; DIS_FILE_NAMES;
DIS_MEM_CAP; WRITELN;
WRITELN('Total lines = ',HIGHEST_NO-1,
', Total characters = ',TOT_CH);
WRITE('Hit any key to continue ');
CH:=READKEY; WRITELN;
END ELSE

```

```

BEGIN
  CLOSE(F); READ_LINES_FROM_FILE(HIGHEST_NO);
  IF (NOT(AVAILABLE_MEMORY)) THEN
  BEGIN WRITELN('File truncated...');WRITELN; END;
  WRITELN('File...',FILE_NAME,' is opened for
    update');
  WRITE('Hit any key to continue ');
  CH:=READKEY; WRITELN;
  DIS_FILE_NAMES; DIS_MEM_CAP; WRITELN;
  WRITELN('Total lines = ',HIGHEST_NO-1,
    ',Total characters loaded = ',TOT_CH);
  WRITE('Hit any key to continue ');
  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...');
  WRITE('Hit any key to continue ');
  CH:=READKEY; WRITELN;
END;
PROCEDURE LOCATE_FILES;
LABEL 10; VAR F_NAME : STRING[12];
BEGIN
  OLD_FILE_NAME:=FILE_NAME;
  REPEAT
    GET_DRIVE_NAME; WRITELN;
    WRITELN('Please enter the file name(8.3 form),
      *,? as wild characters...');
    WRITE('Hit just Enter to abort');READLN(F_NAME);
    IF (F_NAME='') THEN BEGIN WRITELN; GOTO 10; END;
    FILE_NAME:=DR_NAME+'\'+'+F_NAME; SPLIT_FILE_NAME;
    TEST_FILE_NAME;
  UNTIL NOT(INVALID_FILE_NAME);
  WRITELN; WRITE('Locate file name = ',file_name,
    '...Hit any key to continue ');
  CH:=READKEY; WRITELN; WRITELN;
  WRITELN('Work is in progress...Please wait till
    Job Over message');
  COMMAND:='/C '+DIR+'+FILE_NAME'+/A/ON/S/P';
  SWAPVECTORS; EXEC(GETENV('COMSPEC'),COMMAND);
  SWAPVECTORS; IF (DOSERROR<>0) THEN LESS_MEMORY;
10:WRITELN;
  WRITE('Job Over...Use DIR/S in DOS Shell(?) for
    details...Hit any key to continue ');
  CH:=READKEY; WRITELN; FILE_NAME:=OLD_FILE_NAME;
END;
PROCEDURE GET_FILES;
BEGIN
  OLD_FILE_NAME:=FILE_NAME;WRITELN;
  WRITELN('Please enter source file(s)
    information');
  GET_CHECK_FILE_NAME2;
  F_NAME1:=ASCIIZ; FSPLIT(F_NAME1,D2,N2,E2);
  WRITELN('Please enter target file(s)
    information');
  GET_CHECK_FILE_NAME2;
  F_NAME2:=ASCIIZ; FSPLIT(F_NAME2,D3,N3,E3);
  FILE_NAME:=OLD_FILE_NAME;
END;
PROCEDURE MOVE_FILES;
LABEL 5,10;
BEGIN
  GET_FILES; IF (F_NAME1=F_NAME2) THEN
  BEGIN
    WRITELN('File(s) cannot be moved on to the
      self'); GOTO 10;
  END;
  WRITELN('Work is in progress...Please wait');
  IF (D2=D3) THEN
  BEGIN F_NAME1:=N2+E2; F_NAME2:=N3+E3; END;
  WRITELN;
  COMMAND:='/C '+COPY+'+F_NAME1+' '+F_NAME2;
  SWAPVECTORS; EXEC(GETENV('COMSPEC'),COMMAND);
  SWAPVECTORS;
  IF (DOSERROR<>0) THEN
  BEGIN
    LESS_MEMORY;WRITELN('Movement aborted');
    GOTO 10;
  END;
  ASCIIZ:=F_NAME2; DISPLAY_DIRECTORY; WRITELN;
  WRITE('Source file(s) are being deleted. Delete ?
    (Y/y for yes) ');
  CH:=UPCASE(READKEY); WRITELN(CH);
  IF (CH='Y') THEN GOTO 5 ELSE GOTO 10;
5:COMMAND:='/C '+DEL+'+F_NAME1+'/P';
  SWAPVECTORS; EXEC(GETENV('COMSPEC'),COMMAND);
  SWAPVECTORS; WRITELN;
  WRITELN('Source file(s) are deleted/not deleted
    as desired...');
  WRITELN; IF (DOSERROR<>0) THEN LESS_MEMORY ELSE
  WRITE('Movement over ');
  10:WRITELN; WRITE('Hit any key to continue ');
  CH:=READKEY; WRITELN;
END;
PROCEDURE BACK_UP_FILES;
LABEL 10;
BEGIN
  GET_FILES; IF (F_NAME1=F_NAME2) THEN
  BEGIN
    WRITELN('File(s) cannot be copied on to the
      self'); GOTO 10;
  END;
  WRITELN('Work is in progress...Please wait');
  IF (D2=D3) THEN
  BEGIN F_NAME1:=N2+E2; F_NAME2:=N3+E3; END;
  WRITELN;
  COMMAND:='/C '+COPY+'+F_NAME1+' '+F_NAME2;
  SWAPVECTORS; EXEC(GETENV('COMSPEC'),COMMAND);
  SWAPVECTORS;
  IF (DOSERROR<>0) THEN LESS_MEMORY ELSE
  BEGIN
    WRITELN; WRITELN('Back-up of the file(s) i.e.
      Copying file(s) over');
    ASCIIZ:=F_NAME2; DISPLAY_DIRECTORY;
  END;
  10:WRITELN; WRITE('Hit any key to continue ');
  CH:=READKEY; WRITELN;
END;
PROCEDURE RENAME_FILES;
LABEL 10;
BEGIN
  GET_FILES;
  IF (F_NAME1=F_NAME2) THEN
  BEGIN
    WRITELN('File(s) cannot be renamed to the
      self'); WRITELN; GOTO 10;
  END;
  WRITELN('Work is in progress...Please wait');
  IF (D2=D3) THEN
  BEGIN F_NAME1:=N2+E2; F_NAME2:=N3+E3; END;
  WRITELN;
  COMMAND:='/C '+REN+'+F_NAME1+' '+F_NAME2;
  SWAPVECTORS;
  EXEC(GETENV('COMSPEC'),COMMAND); SWAPVECTORS;
  IF (DOSERROR<>0) THEN LESS_MEMORY ELSE
  BEGIN
    ASCIIZ:=F_NAME2;
    WRITELN('Files are renamed as desired.
      Check...'); DISPLAY_DIRECTORY;
  END;
  10:WRITELN; WRITE('Hit any key to continue ');
  CH:=READKEY; WRITELN;
END;
PROCEDURE DOS_SHELL;
BEGIN
  WRITELN; WRITELN('Enter executable file name with
    path and parameters...');
  WRITELN('e.g. tcc -If:\tc\include -Lf:\tc\lib
    nceditor.c to compile a ',
    'nceditor.c programfrom the f:\tc\bin
    directory or tpc npeditor.pas to ',
    'compile a npeditor.pas');
  WRITELN('program from the f:\pascal5.5
    directory');
  WRITELN('Or any internal command with
    parameters(Enter EXIT to return)');
  WRITELN; SWAPVECTORS;
  EXEC(GETENV('COMSPEC'),' /C COMMAND.COM');
  SWAPVECTORS; IF (DOSERROR<>0) THEN LESS_MEMORY;
END;

```

```

BEGIN{MAIN 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; TEXTCOLOR(YELLOW);
FOR I:=1 TO 50 DO CLEAR1[I]:=CHR(ORD(10));
NEW_FILE;
REPEAT
BEGIN
1:WRITELN; DATE_TIME;
WRITE('File:',FILE_NAME,' ','Date:',DAY,'/',
MONTH,'/',YEAR,' ','Day:',DAY_NAM);
WRITE(' ','Time:');
WRITELN(HOURS,':',MINUTES,':',SECONDS);
WRITE('-----',
'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-Entirelist/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/B-Backup/',
'X-Move files/Z-Rename files/@-Locate
files/?-DOS shell/',
'N-Open new file/Q-Quit(Upper/Lower Case) ');
OPTION:=UPCASE(READKEY); WRITELN(OPTION);
WRITELN;
IF ((OPTION<'?') OR (OPTION>'Z') OR
(OPTION=CHR(0))) THEN
BEGIN WRITELN('Retry...'); WRITELN; 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 ');
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 ');
CH:=READKEY; WRITELN;
END;
'I':BEGIN
WRITELN('Insert lines...I/i'); WRITELN;
WRITE('Before which line number ? ');
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 ');
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 ');
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) ');
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 ');
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 ');
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? ');
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 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;
WRITELN;
10:

```

```

WRITE('Total lines = ',HIGHEST_NO-1,
      ', Copying & Pasting over...',
      'Hit any key to continue ');
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 ?');
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;
WRITE('Cut & Paste over...Hit any key to
continue '); 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 ? ');
READLN(A); VAL(A,NUMBER,CODE);
NUMBER_CHECK(NUMBER); WRITELN;
IF (NUMBER<1) THEN NUMBER:=1;
IF (NOT(NUMBER>HIGHEST_NO-1)) THEN
BEGIN
WRITELN('You can have up to 132 characters
in the line. ',
'End the line with Enter key.');
```

```

INSERT_LINE(NUMBER,NEW_LINE);
IF (NOT(AVAILABLE_MEMORY)) THEN GOTO 20;
WRITELN; WRITELN;
WRITE('Updation over...Hit any key to
continue '); CH:=READKEY;
END ELSE
BEGIN
WRITELN('Invalid number...Maximum line
number existing is...',HIGHEST_NO-1);
WRITELN; WRITE('Updation aborted...Hit any
key to continue ');
CH:=READKEY;
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) ');
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) '); 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');
```

```

CH:=READKEY; WRITELN;
END;
REPEAT
FVAL:=I+NUMBER-1; WRITELN(CLEAR1);
IF (FVAL>=STOP) THEN FVAL:=STOP;
LIST_LINES(I,FVAL); I:=FVAL+1; WRITELN;
WRITE('Enter character S/s to stop or any
other character to continue ');
CH:=UPCASE(READKEY);
UNTIL ((FVAL=STOP) OR (CH='S'));
WRITELN; WRITELN;
WRITE('Display over...Hit any key to
continue '); 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; DIS_OPEN_FILE; 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
END ELSE
```

```

BEGIN
  WRITELN;
  WRITELN('String length = 0...So no
    searching');
  WRITELN;WRITE('Hit any key to continue');
  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) ');
      WHILE (NOT EOLN) DO
      BEGIN M:=M+1; READ(REP_STRING[M]); END;
      READLN;
      IF (M>20) THEN
      BEGIN
        WRITELN('Number of characters > 20,First
          20 characters are considered');
        M:=20;
      END;
      FOR I:=1 TO M DO
      TEMP12[I]:=REP_STRING[I];
      B13:=TEMP12; WRITELN;
      WRITE('Replace by query?(Y/y for yes) ');
      QUERY:=UPCASE(READKEY); WRITELN(QUERY);
      NUMBER:=1; STOP:=HIGHEST_NO-1; WRITELN;
      REPLACE_STRING(NUMBER,STOP,B12,B13);
      END ELSE
      BEGIN
        WRITELN;
        WRITELN('String length = 0...So no
          replacement');
        WRITE('Hit any key to continue ');
        CH:=READKEY; WRITELN;
        END; DIS_MEM_CAP; SAVED:=FALSE;
      END ELSE ZERO_LINES;
    END;
  END;
'T':BEGIN
  WRITELN('Copy & Paste lines from another
    file...T/t');
  OLD_FILE_NAME:=FILE_NAME;
  GET_CHECK_FILE_NAME1;
  FINDFIRST(FILE_NAME,ANYFILE,DIRINFO);
  IF (DOSERROR=0) THEN
  BEGIN
    WRITE('Paste before which line number?');
    READLN(A); VAL(A,NUMBER,CODE);
    NUMBER_CHECK(NUMBER);
    IF (NUMBER<1) THEN NUMBER:=1;
    IF (NUMBER>HIGHEST_NO) THEN
    BEGIN
      INSERT_REQUIRED_BLANK_LINES(NUMBER);
      IF (NOT(AVAILABLE_MEMORY)) THEN GOTO 25;
    END;
    READ_LINES_FROM_FILE(NUMBER);
    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: FILE_NAME:=OLD_FILE_NAME; WRITELN;
  SAVED:=FALSE;
  WRITE('Hit any key to continue ');
  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; DIS_OPEN_FILE;
    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; WRITELN;
  WRITE('Hit any key to continue ');
  CH:=READKEY; WRITELN;
  END;
'J':BEGIN
  WRITELN('Search archive file(s) in a
    directory for a string ...J/j');
  OLD_FILE_NAME:=FILE_NAME; 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; 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 ');
    CHOICE:=UPCASE(READKEY); WRITE(CHOICE);
    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 ');
      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 ');
      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;
  WRITE('Hit any key to check ');
  CH:=READKEY; WRITELN;
  DISPLAY_DIRECTORY;
  FILE_NAME:=OLD_FILE_NAME; WRITELN;
  DIS_OPEN_FILE;
  END;

```



```

'Z':BEGIN
  WRITELN('Rename an archive file/directory
           or files/directories...Z/z');
  RENAME_FILES;
END;
'X':BEGIN
  WRITELN('Move one or more archive
           files...X/x');
  MOVE_FILES;
END;
'B':BEGIN
  WRITELN('Back-up archive file(s)...B/b');
  BACK_UP_FILES;
END;
'@':BEGIN
  WRITELN('Locate archive file(s)...@');
  LOCATE_FILES;
END;
'?':BEGIN
  WRITELN('DOS shell...?'); DOS_SHELL;
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
    WRITELN('Text is possibly changed but not
             saved');

    WRITELN;
    WRITE('Do you want to save it(write in a
           file) ? (Y/y for yes) ');
    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
    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;
    WRITE('See you again...Bye...Hit any
           key');
    CH:=READKEY; WRITELN;
    TEXTBACKGROUND(BLACK); TEXTCOLOR(WHITE);
    HALT;
  END;
  END;{END OF CASE Q/q}
END; {END OF CASE}
END; {END OF REPEAT}
UNTIL ((OPTION='Q') AND (SAVE<>'Y'));
END.{MAIN ENDS}

```