

```

10 (Program to list lines of a file in reverse order.
20 Shows use of pointers, records, MARK and RELEASE.)
30
40 PROGRAM ReverseLine;
50
60 TYPE elem=RECORD                                (Create linked-list structure)
70     next: ^elem;
80     ch: CHAR
90     END;
100 link:=^elem;
110
120 VAR prev,cur,heap: link;                        (Call pointers to ^elem)
130
140 BEGIN
150 REPEAT                                          (do this many times)
160     MARK(heap);                                (assign top of heap to ^heap)
170     prev:=NIL;                                  (points to no variable yet)
180     WHILE NOT EOLN DO
190         BEGIN
200             NEW(cur);                            (create a new dynamic record)
210             READ(cur^.ch);                       (and assign its field to one
220                 character from file.)
230             cur^.next:=prev;                     (this field's pointer addresses)
240             prev:=cur                             (previous record.)
250         END;
260
270 (Write out the line backwards by scanning the records
280 set up backwards.)
290
300     cur:=prev;
310     WHILE cur <> NIL DO                          (NIL is first)
320         BEGIN
330             WRITE(cur^.ch);                       (WRITE this field i.e. character)
340             cur:=cur^.next                         (Address previous field.)
350         END;
360     WRITELN;
370     RELEASE(heap);                                (Release dynamic variable space)
380     READLN                                         (Wait for another line)
390 UNTIL FALSE                                       (Use CC to exit)
400 END.

```