

Control section	Symbol name	Address	Length
PROGA		4000	0063
	LISTA	4040	
	ENDA	4054	
PROGB		4063	007F
	LISTB	40C3	
	ENDB	40D3	
PROGC		40E2	0051
	LISTC	4112	
	ENDC	4124	

Pass 1:

```

begin
  get PROGADDR from operating system
  set CSADDR to PROGADDR {for first control section}
  while not end of input do
    begin
      read next input record {Header record for control section}
      set CSLTH to control section length
      search ESTAB for control section name
      if found then
        set error flag {duplicate external symbol}
      else
        enter control section name into ESTAB with value CSADDR
      while record type ≠ 'E' do
        begin
          read next input record
          if record type = 'D' then
            for each symbol in the record do
              begin
                search ESTAB for symbol name
                if found then
                  set error flag {duplicate external symbol}
                else
                  enter symbol into ESTAB with value
                    (CSADDR + indicated address)
                end {for}
              end {while ≠ 'E'}
            add CSLTH to CSADDR {starting address for next control section}
          end {while not EOF}
        end {Pass 1}

```

Figure 3.13(a) Algorithm for Pass 1 of a linking loader.

Pass 2:

```

begin
  set CSADDR to PROGADDR
  set EXECADDR to PROGADDR
  while not end of input do
    begin
      read next input record {Header record}
      set CSLTH to control section length
      while record type ≠ 'E' do
        begin
          read next input record
          if record type = 'T' then
            begin
              {if object code is in character form, convert
                into internal representation}
              move object code from record to location
                (CSADDR + specified address)
            end {if 'T'}
          else if record type = 'M' then
            begin
              search ESTAB for modifying symbol name
              if found then
                add or subtract symbol value at location
                  (CSADDR + specified address)
              else
                set error flag {undefined external symbol}
              end {if 'M'}
            end {while ≠ 'E'}
          if an address is specified {in End record} then
            set EXECADDR to (CSADDR + specified address)
          add CSLTH to CSADDR
        end {while not EOF}
      jump to location given by EXECADDR {to start execution of loaded program}
    end {Pass 2}

```

Figure 3.13(b) Algorithm for Pass 2 of a linking loader.

Pass 2 of our loader [Fig. 3.13(b)] performs the actual loading, relocation, and linking of the program. CSADDR is used in the same way it was in Pass 1—it always contains the actual starting address of the control section currently being loaded. As each Text record is read, the object code is moved to the specified address (plus the current value of CSADDR). When a Modification record is encountered, the symbol whose value is to be used for modification is looked up in ESTAB. This value is then added to or subtracted from the indicated location in memory.

The last step performed by the loader is usually the transferring of control to the loaded program to begin execution. (On some systems, the address