Esercizio 1

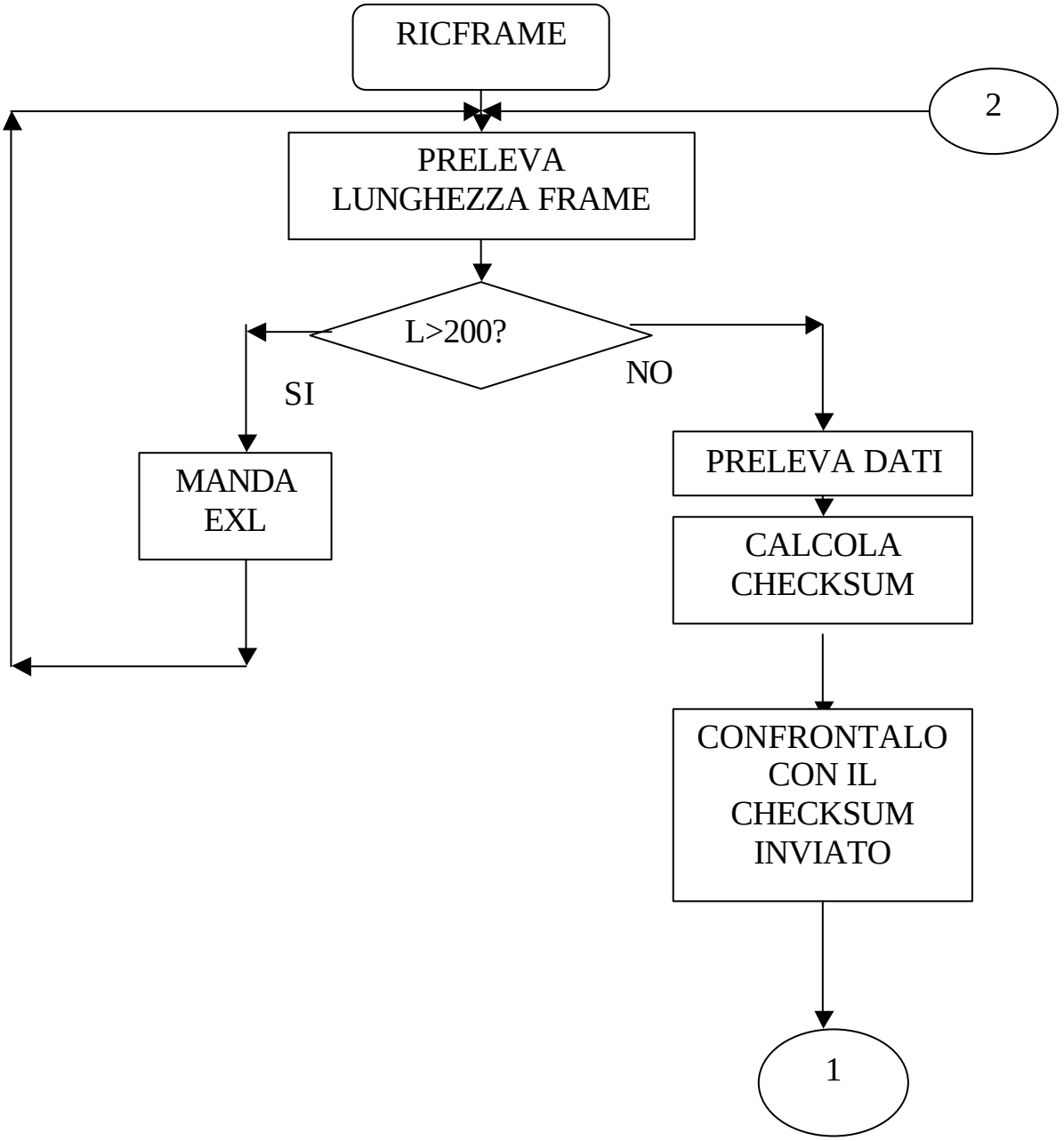
Traccia

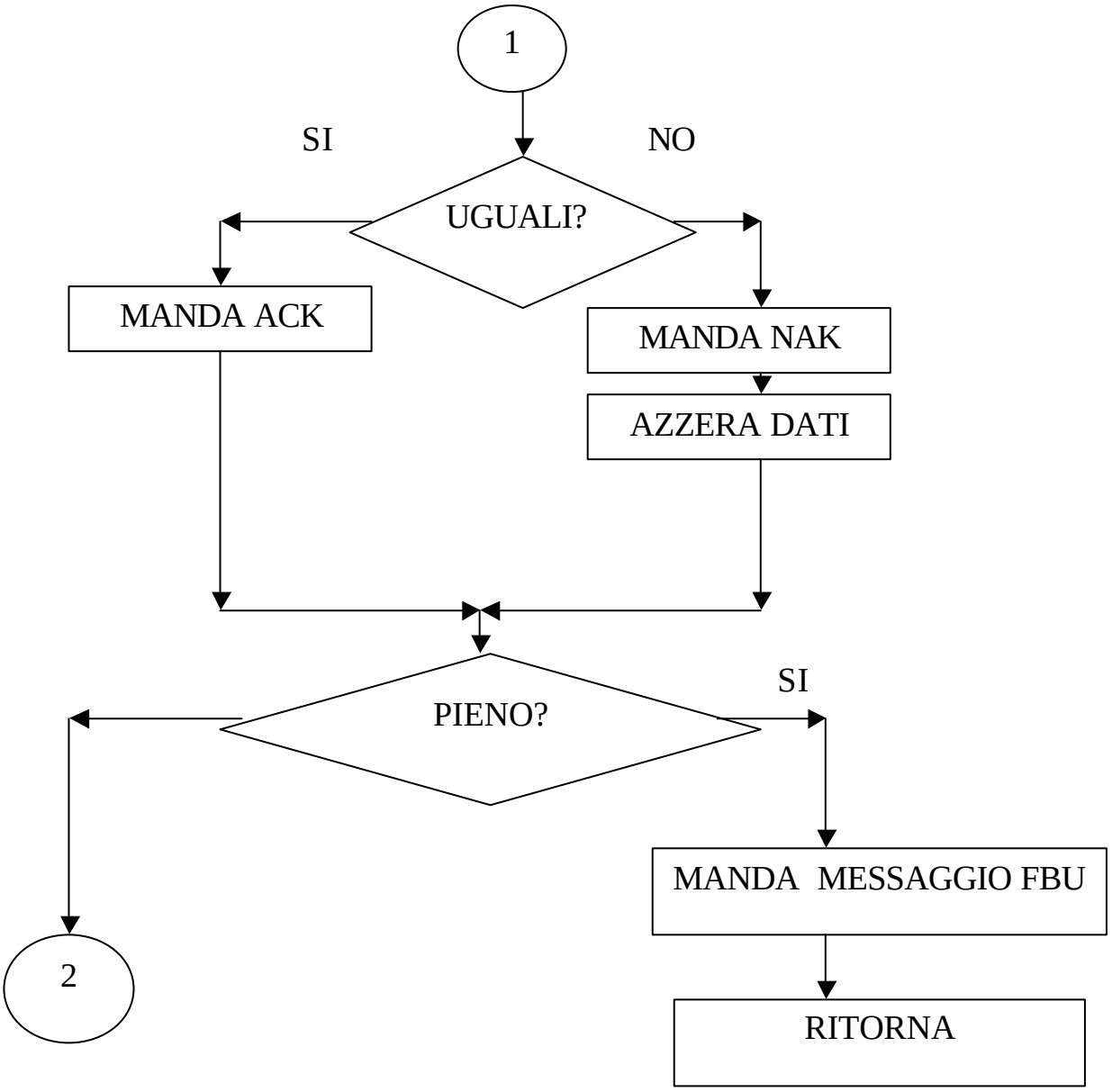
IL microprocessore riceve dati dalla porta di indirizzo 30H. I dati sono organizzati in frame nel seguente modo: il primo byte dice la lunghezza complessiva del frame, seguono i byte di dati, l'ultimo byte è il byte di checksum. Esso è un byte di controllo per controllare in ricezione se vi sono state alterazioni dei dati. Tale byte viene realizzato dalla macchina mittente nel modo seguente: si fa prima la XOR fra 00H e il primo byte di dati, poi si fa la XOR fra il risultato precedente e il secondo byte e così via fino all'ultimo byte (vedi simulazione). La lunghezza del frame nel sistema di comunicazione in cui è inserito il nostro microprocessore, può essere al massimo di 200 byte. Lo Z80 preleva il primo byte e controlla che la lunghezza sia superiore a quella consentita. In tal caso invia alla porta un messaggio Excessive Lenght (EXL) pari ad $A3_H$, altrimenti immagazzina i dati in memoria escluso il byte iniziale e il byte di checksum. Successivamente il microprocessore ricalcala il checksum sui dati contenuti nel frame e lo confronta con quello ricevuto. Se i due checksum sono

uguali, lo Z80 invia alla porta un byte di Acknowledgment (ACK), il cui valore è FF_H altrimenti invia un messaggio Not Acknowledgment (NAK) il cui valore è FE_H e cancella l'area di memoria in cui aveva registrato il frame. Successivamente passa ad attendere il nuovo frame o una copia di quello ricevuto con errori e scartato. L'area di memoria utilizzata per immagazzinare i dati è costituita da 10000 locazioni. Quando questa area si riempie il microprocessore invia un messaggio Full Buffer (FBU) pari a 10H alla porta per impedirle di inviare nuovi frame. Scrivere il sottoprogramma RICFRAME che gestisce questo processo. Il sottoprogramma termina quando il buffer è pieno.

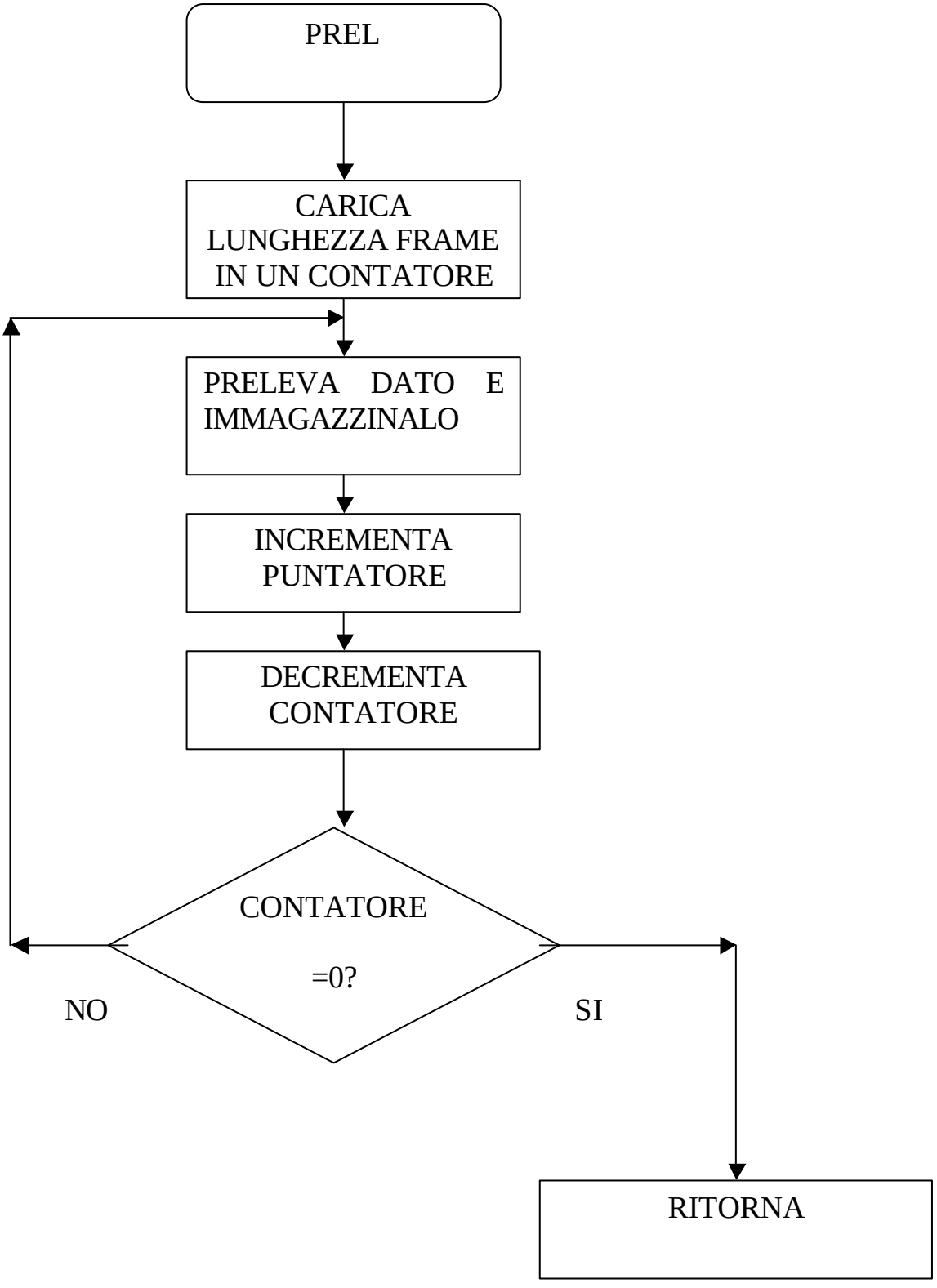
Analisi

Cominciamo una prima stesura del programma mediante una flow chart

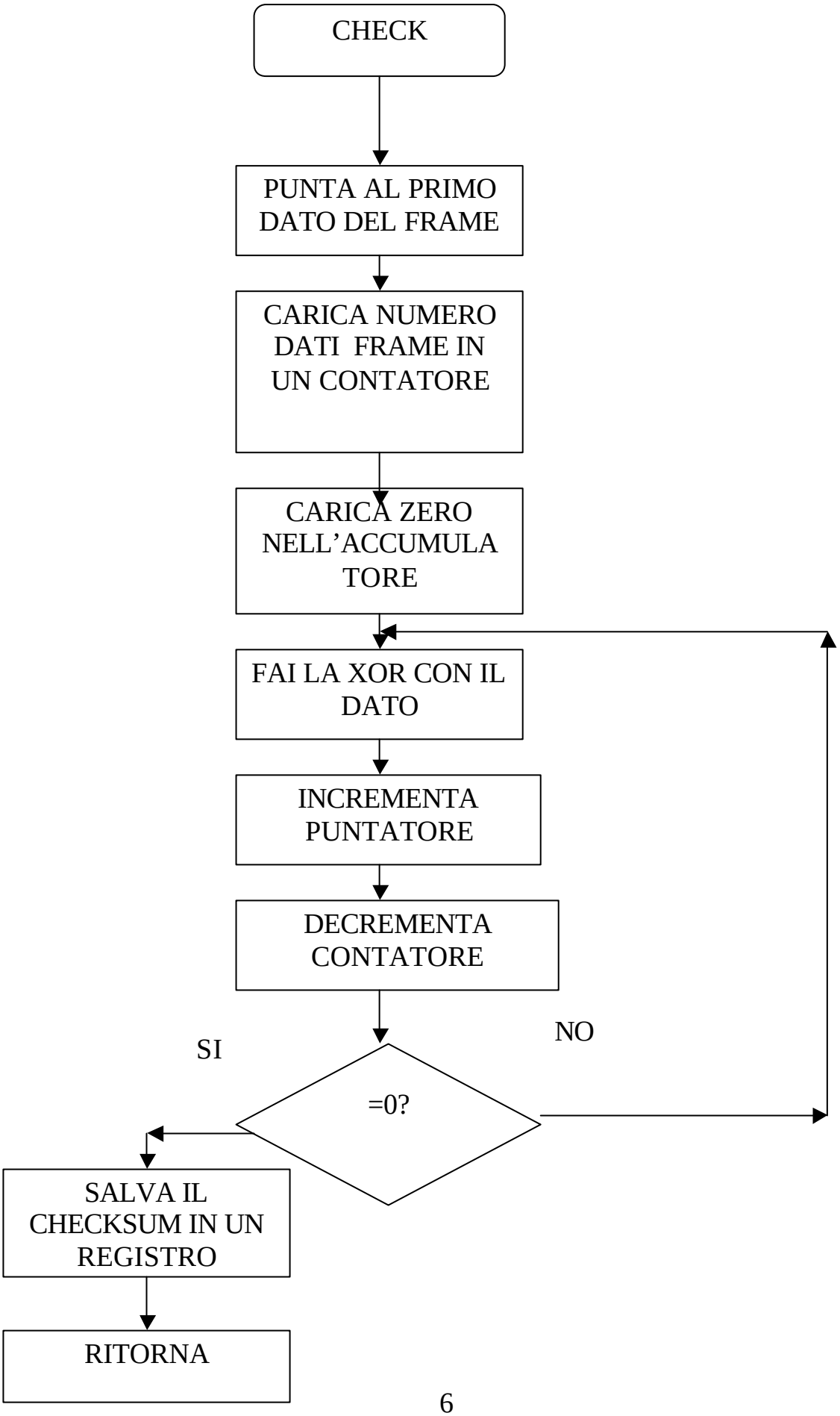

RICFRAME
2
PRELEVA
LUNGHEZZA FRAME
L>200?
SI
NO
MANDA
EXL
PRELEVA DATI
CALCOLA
CHECKSUM
CONFRONTALO
CON IL
CHECKSUM
INVIATO
1



Il blocco preleva dati può essere realizzato con un altro sottoprogramma PREL che può essere rappresentato dal seguente diagramma



Il blocco calcola checksum può essere realizzato con un altro sottoprogramma CHECK


CHECK
PUNTA AL PRIMO DATO DEL FRAME
CARICA NUMERO DATI FRAME IN UN CONTATORE
CARICA ZERO NELL'ACCUMULATORE
FAI LA XOR CON IL DATO
INCREMENTA PUNTATORE
DECREMENTA CONTATORE
NO
SI
=0?
SALVA IL CHECKSUM IN UN REGISTRO
RITORNA

Codifica programma

```
RICFRAME: LD HL,1A00H; punta all'inizio del buffer
ET:       IN A,(30h); leggi la lunghezza del frame contenuta nel primo byte
          CP C9h;controlla se la lunghezza è superiore a 200, pari a C8 hex
          JP N,ET1;se la lunghezza del frame è inferiore salta le istruzioni
                  ;che inviano il messaggio EXL
          LD A,A3h; carica in A il codice di EXL
          OUT (30h),A;invialo alla porta
          JP ET;torna all'inizio
ET1:      CALL PREL; se la lunghezza è giusta preleva i dati,
                   ;il sottoprogramma deve lasciare A inalterato poiché
                   ; la lunghezza del frame
          CALL CHECK;al termine del sottoprogramma HL punterà
                    ; alla locazione che contiene il checksum inviato,
                    ; imponiamo che lasci in D il checksum
                    ;calcolato
         LD E,A;salva A che contiene la lunghezza del frame
         LD A,D;poni in A il checksum calcolato
         CP (HL);confronta con il checksum calcolato
         JP Z,ET2;salta le istruzioni che mandano il messaggio NAK
          LD A,FEh; inserisci in A il valore di NAK
         OUT(30h),A
```

```
        LD A,E; recupera la lunghezza del frame
        CALL AZZERA
        JP ET3;occorre saltare le istruzioni da eseguire se il checksum
              ; è corretto
ET2:    LD A,FFh ;carica in A il codice ACK
        OUT(30h),A
ET3:    AND A; questa istruzione non modifica l'accumulatore ma serve
              ;soltanto per azzerare il carry
        LD BC,410Fh;il buffer inizia con la locazione 1A00, e finisce alla
                   ; 410Fh poiché 10000 in esadecimale è 2710h da cui
                   ; la locazione finale è 1A00+2710-1=410Fh
                   ; il sottoprogramma CHECK dovrà fare in modo che HL
                    ; punti al checksum dell'ultimo frame inserito quindi , se
                   ; è l'ultimo frame inserito HL dovrà contenere 410F
        SBC HL,BC;questa è l'unica istruzione di sottrazione a 16 bit,
                  ;perciò abbiamo azzerato il carry
        JP Z,FINE;in questo caso vai alla fine del sottoprogramma
        ADD HL,BC;altrimenti ripristina HL
        INC HL;incrementalo in modo da puntare alla prima locazione vuota
        JP ET; e ritorna all'inizio per prelevare un nuovo frame
FINE:   LD A,FEh;invia il messaggio di buffer pieno
        OUT(30h),A
```

```
        RET
PREL:   LD B,A;salva nel contatore B la lunghezza del frame
            ; HL contiene già l'indirizzo della prima locazione del buffer
           : da riempire
        INIR
        RET
CHECK:  LD C,A
        LD B,00h;abbiamo inserito nel registro BC la lunghezza del frame
        AND A;azzera il carry
        SBC HL,BC;il sottoprogramma PREL ha portato HL più avanti di una
                ; posizione per cui se sottraggo la lunghezza del frame
                ;mi porto al primo byte del frame
        INC HL;mi devo portare al secondo byte che è il primo dei dati
        LD B,C;uso B come contatore
        DEC B;deve essere pari alla lunghezza del frame meno uno
            ;poiché non dovremo fare la XOR anche con il byte di checksum
            ;inviato dal trasmettitore
        LD E,A;salvo in E la lunghezza del frame
        LD A,00h;A conterrà il checksum
ET4:    XOR (HL)
        INC HL
        DJNZ ET4
```

```
            LD D,A ;salva il checksum calcolato
            LD A,E;salva di nuovo in A la lunghezza del frame
            RET
AZZERA:  LD B,A;il sottoprogramma CHECK ha fatto in modo che HL punti
                ;all'ultimo byte del frame
            LD A,00h
ET5:        LD (HL),A
            DEC HL
            DJNZ ET5
            INC HL
            RET
```