```
0A1B  87  GLO   R7 ;D=R7.0 (current i)
  1C  52  STR   R2 ;Push for comparing
  1D  9F  GHI   RF ;D=RF.1 (last card address)
  1E  F3  XOR      ;Compare R7.0:RF.1
  1F  3A  BNZ      ;If ≠, branch to 0A11

0A20  11
  21  8E  GLO   RE
  22  32  BZ       ;Branch if =0 to 0A31 (no matches)
  23  31
  24  FC  ADI      ;Else add 1 to value
  25  01
  26  FE  SHL      ;Else move count to high 4 bits
  27  FE  SHL      ;by shifting left four times
  28  FE  SHL
  29  FE  SHL
  2A  52  STR   R2 ;Push for "OR"ing
  2B  0A  LDN   RA ;D=M(R(A)) Get card N (which has matches)
  2C  FA  ANI      ;"AND" with 0F to strip suit
  2D  0F
  2E  F1  OR       ;"OR" with count on stack (byte packed)
  2F  5C  STR   RC ;M(R(C))=D  Store in pairs area

0A30  1C  INC   RC ;RC=RC+1
  31  87  GLO   R7
  32  AA  PLO   RA ;RA.0=R7.0  (advance to next unequal card)
  33  52  STR   R2 ;but push value for comparing
  34  9F  GHI   RF ;D=RF.1 (get last card address)
  35  F3  XOR      ;Compare RA.0:RF.1
  36  3A  BNZ      ;If ≠, branch to 0A0E  (not done)
  37  0E
  38  F8  LDI      ;Else store stop byte after pairs (or no pairs)
  39  FF
  3A  5C  STR   RC ;M(R(C))=FF
  3B  1C  INC   RC ;RC=RC+1  (RC=TA now)

                      (RECOMMEND THROW OUTS)

0A3C  9A  GHI   RA ;D=RA.1  (hand high order address)
  3D  BD  PHI   RD ;RD.1=D  (RD = pairs pointer for this test)
  3E  9F  GHI   RF ;D=RF.1  (get address of last card)
  3F  FF  SMI      ;D=D-04  (subtract 04)

0A40  04
  41  AA  PLO   RA ;RA.0=D  (reset RA to first card)
  42  8F  GLO   RF ;D=RF.0  (get saved pairs address)
  43  AD  PLO   RD ;RD.0=D  (RD = pairs pointer)
  44  0D  LDN   RD ;D=M(R(D))  (get a possible pair value)
  45  FB  XRI      ;           (but test if = FF  stop byte)
```