

```

0A70 FB XRI      ;Compare with FF
71  FF
72 32 BZ        ;If=FF, branch to 0A79 (there is only one pair)
73 79
74 0D LDN      RD ;D=M(R(D)) (get same definite pair)
75 F6 SHR      ;Shift right x 4 to get only the pair's number
76 F6 SHR      ;      "      "      "
77 F6 SHR      ;      "      "      "
78 F6 SHR      ;      "      "      "
79 52 STR      R2 ;Push (J1) (either a pair # or a zero byte)
7A 8E GLO      RE ;D=RE.0 (get J value)
7B F5 SD        ;Subtract M(R(X))-D (J1-J)
7C 33 BPZ      ;If J1 ≥ J then branch to 0A80 (D holds correct value)
7D 80
7E 8E GLO      RE ;D=RE.0 else get same value of J
7F F7 SM        ;Subtract D-M(R(X)) subtract other way around

0A80 BE PHI      RE ;RE.1=D (save the embryo value)
81 30 BR        ;Branch to 0AC4 (final decode)
82 C4

```

(STRAIGHTS)

```

0A83 F8 LDI
84 00
85 BE PHI      RE ;RE.1=0 (to initialize)
86 9F GHI      RF ;D=RF.1 (get last card address)
87 FF SMI      ;Subtract 04
88 04
89 AA PLO      RA ;RA.0=D (RA points to first card)
8A 0A LDN      RA ;D=M(R(A)) (get first card)
8B FA ANI      ;"AND" with 0F to strip suit
8C 0F
8D 52 STR      R2 ;Push for sequential comparison
8E F0 LDX      ;Pop value off stack
8F FC ADI      ;Add 01

0A90 01
91 52 STR      R2 ;Push new value
92 1A INC      RA ;RA=RA+1 (next card)
93 0A LDN      RA ;D=M(R(A)) (get next card)
94 FA ANI      ;"AND" with 0F to strip suit
95 0F
96 F3 XOR      ;Compare with byte on stack
97 3A BNZ      ;If ≠, then not a straight, branch to 0AA4
98 A4          (go test flush)
99 22 DEC      R2 ;Decrement stack to preserve value
9A 8A GLO      RA ;D=RA.0
9B 52 STR      R2 ;Push for comparing

```