

```

TO suchen1 ivorrat :gelegt ivorbei
  LOCAL "karte "
  IF EQUALP 9 COUNT :gelegt * >
    [loesung :gelegt 1]
  IF EQUALP 0 COUNT ivorrat * >
    [STOP]
  MAKE "karte FIRST ivorrat *
  REPEAT (IF EQUALP 4 COUNT :gelegt [1] [4] " ) *
    [( IF testkarte :gelegt ikarte " *
      [suchen SE ivorbei BF ivorrat LPUT ikarte :gelegt [ ] " ] ) *
      MAKE "karte drehe ikarte " ]
    suchen BF ivorrat :gelegt LPUT FIRST ivorrat ivorbei "
END

```

```

TO testkarte1 :gelegt ikarte
  IF EMPTYP :gelegt " >
    [OP "TRUE]
  IF NOT EQUALP 0 mod COUNT :gelegt 3 " >
    [(IF NOT test2 LAST LAST ikarte ITEM 2 LAST LAST :gelegt " >
      [OP "FALSE]]
  IF (COUNT :gelegt) > 2 " >
    [(IF NOT test2 FIRST LAST ikarte *
      ITEM 3 LAST ITEM ((COUNT :gelegt) - 2) :gelegt " >
      [OP "FALSE]]
    [OP "TRUE "
END

```

```

TO test21 ih1 ih2
  IF NOT EQUALP FIRST ih1 FIRST ih2 *
    [OP "FALSE] *
    [OP NOT EQUALP LAST ih1 LAST ih2]
END

```

```

TO mod1 ia ib
  IF ia < ib *
    [OP ia] *
    [OP mod (ia - ib) ib]
END

```

```

TO drehe1 :karte
  OP LPUT SE BF LAST :karte FIRST LAST :karte *
  DE FIRST :karte mod (1 + ITEM 2 :karte) 4
END

```

```

TO loesung1 :l
  PR [ ]
  PR [eine Loesung ist:]
  prloesung :l
  PR [ ]
END

```

```

TO prloesung1 :liste
  IF EMPTYP :liste [STOP]
  PR FIRST :liste
  prloesung BF :liste
END

```