

Hersheyn kirjaimet

Jarmo Hietala

2018-12-08

Hersheyn kirjaimet

Hersheyn kirjaimet ovat kokoelma vektorikirjasimia vuodelta 1967. Se on yksi ensimmäisiä vektorimuotoisia kirjasimia. Kokoelma on vapaasti jaettavissa.

Hersheyn kirjaimet julkaisi Allen V. Hershey (1910–2004) teoksessa *Calligraphy for Computers* (Hershey 1967). Hershey oli koulutukseltaan fyysikko ja työskenteli pääosan elämästään Yhdysvaltain laivastossa.

Hersheyn kirjaimet koostuvat ruudukkoon piirretyistä suorista viivoista. Niiden alkuperäisessä käyttötarkoituksessaan viivat piirrettiin katodisädeputkella valokuvapaperille. Valokuvapaperi siirrettiin tämän jälkeen painokoneelle. Tietokonekuvaruutuja ei tuon ajan tietokoneissa juurikaan ollut, vaan ensisijainen tulostusväline oli paperi.

Levityksessä olevat kirjasimet koostuvat kahdesta kokoelmasta: länsimaisesta (occidental) ja itämaisesta (oriental). Länsimaisen kokoelman tiedostonniminä ovat `hersh.oc1`, `hersh.oc2`, `hersh.oc3` ja `hersh.oc4`, ja itämaisien kokoelman tiedostonniminä `hersh.or1`, `hersh.or2`, `hersh.or3` ja `hersh.or4`.

Kukin tiedosto esittää merkkien kärkipistekoordinaatit kirjainpareina seuraavaan tapaan (merkit 1–6):

```
1 9MWRMNV RRMVV RPSTS
2 16MWOMOV ROMSMUNUPSQ ROQSQRUUSVOV
3 11MXVNTMRMPNOPOSPURVTVVU
4 12MWOMOV ROMRMTNUPUSTURVOV
5 12MWOMOV ROMUM ROQSQ ROVUV
6 9MVOMOV ROMUM ROQSQ
```

...

Rivin viisi ensimmäistä merkkiä (välilyönteineen) kertoo kirjaimen järjestysluvun, kolme seuraavaa kärkipisteiden lukumäärän. Kärkipistekoordinaatit ovat suhteessa R-kirjaimen järjestyslukuun: R-kirjain on nolla, R-kirjainta edeltävät kirjaimet negatiivisia ja R-kirjaimen jälkeiset kirjaimet positiivisia koordinaatteja. Koordinaatiston nollapiste sijaitsee merkin keskellä. Ensimmäisessä koordinaattiparissa on vasemman ja oikean laidan sijainnit. Tämän jälkeen tulevat kärkipisteiden koordinaatit (x,y)-kirjainpareina muodostaen yhtenäisiä viivoja. Merkkipari `␣R` tarkoittaa kynän siirtämistä uuteen sijaintiin viivaa piirtämättä.

Hersheyn kirjainten länsimäisessä kokoelmassa on noin 1500 merkkiä. Kokoelma sisältää latinalaisen aakkoston useina erilaisina versioina, kreikkalaiset aakkoset, kyrilliset kirjaimet ja erilaisia muun muassa kartografisia ja matemaattisia symboleja. Ä- ja Ö-kirjainta kokoelmiin ei kuulu.

Itämainen kokoelma koostuu noin 800 japanilaisesta kanji-, hiragana- ja katakana-merkistä.

Toteutus Haskell-kielellä

Käytännön kannalta helpoin toteutustapa on koota tarvittavien merkkien kärkipistekoordinaatit omaksi kirjastokseen. Teemme sen seuraavassa Haskell-kielellä:

```
import Data.Char (ord)
import Data.List (intercalate)
import Data.List.Split (splitWhen)
import qualified Data.Map as Map

fst s xs = [a | (a,b) <- xs]
snd s xs = [b | (a,b) <- xs]

pairs (x:y:xs) = xy1 : pairs xs
  where
    xy1 = (ord x - ord 'R', ord y - ord 'R')
pairs _ = []

hershey xs = [[gr]] ++ splitWhen (==( -50,0)) zs
  where
    zs = pairs x3
    x3 = drop 2 x2
    gr = (leftHandPos, rightHandPos)
    [(leftHandPos, rightHandPos)] = pairs (take 2 x2)
    x2 = drop 8 xs

startsNew xs = all (`elem` "0123456789 ") ys
  && length ys == 8
  where
    ys = take 8 xs

process (x:y:xs)
  | startsNew y = x : process (y:xs)
  | otherwise  = process ((x ++ y) : xs)
process xs = xs

show1 xy = "module Hershey\n(\n  hershey1\n)\nwhere\n\n" ++
  "hershey1 = [\n " ++ x1 ++ "\n ]"
  where
    x1 = intercalate ",\n " x2
    x2 = map show xy

cut n (x:y:xs)
```

```

    | n > 48 && (x==' ' || x==']') && y==' ' && xs /= [] =
      [x] ++ [y] ++ "\n      " ++ cut 6 xs
    | otherwise = x : cut (n+1) (y:xs)
cut n (y:xs) = [y]
cut _ _ = []

```

```

mkModuleText content = x5
  where
    ls1 = process (lines content)
    uppers = zip ['A'..'Z'] (drop 89 ls1)
    lowers = zip ['a'..'z'] (drop 166 ls1)
    numbers = zip ['0'..'9'] (drop 251 ls1)
    punctuation = zip [' ', '-', '.', ', ', ': ', '; ', '!', '?']
      (ls1 !! 250 : ls1 !! 275 : drop 261 ls1)
    letnum = uppers ++ lowers ++ numbers ++ punctuation
    map1 = Map.fromList letnum
    ltrs = fsts letnum
    cx = [map1 Map.! c | c <- ltrs]
    xyss = map hershey cx
    xys = zip ltrs xyss
    x2 = show1 xys
    x3 = lines x2
    x4 = map (cut 0) x3
    x5 = unlines x4

```

```

main = do
  let hersheyFile = "grass/hersheys.oc"
  content <- readFile hersheyFile
  let mdt = mkModuleText content
  putStrLn mdt

```

Funktio `mkModuleText` tulostaa nyt listan kärkipistekoordinaateista. Tallennamme listan kirjastotiedostoon `Hershey.hs`.

```

module Hershey
(
  hershey2
)
where

```

```

hershey2 = [
  ('A', [[(-9,9)], [(0,-12), (-8,9)], [(0,-12), (8,9)], [(-5,2),
    (5,2)]]),
  ('B', [[(-11,10)], [(-7,-12), (-7,9)], [(-7,-12), (2,-12),

```

$(5, -11), (6, -10), (7, -8), (7, -6), (6, -4), (5, -3), (2, -2)],$
 $[(-7, -2), (2, -2), (5, -1), (6, 0), (7, 2), (7, 5), (6, 7),$
 $(5, 8), (2, 9), (-7, 9)]]),$
('C', $[[(-10, 11)], [(8, -7), (7, -9), (5, -11), (3, -12), (-1, -12),$
 $(-3, -11), (-5, -9), (-6, -7), (-7, -4), (-7, 1), (-6, 4),$
 $(-5, 6), (-3, 8), (-1, 9), (3, 9), (5, 8), (7, 6), (8, 4)]]),$
('D', $[[(-11, 10)], [(-7, -12), (-7, 9)], [(-7, -12), (0, -12),$
 $(3, -11), (5, -9), (6, -7), (7, -4), (7, 1), (6, 4), (5, 6),$
 $(3, 8), (0, 9), (-7, 9)]]),$
('E', $[[(-10, 9)], [(-6, -12), (-6, 9)], [(-6, -12), (7, -12)],$
 $[(-6, -2), (2, -2)], [(-6, 9), (7, 9)]]),$
('F', $[[(-10, 8)], [(-6, -12), (-6, 9)], [(-6, -12), (7, -12)],$
 $[(-6, -2), (2, -2)]]),$
('G', $[[(-10, 11)], [(8, -7), (7, -9), (5, -11), (3, -12), (-1, -12),$
 $(-3, -11), (-5, -9), (-6, -7), (-7, -4), (-7, 1), (-6, 4),$
 $(-5, 6), (-3, 8), (-1, 9), (3, 9), (5, 8), (7, 6), (8, 4),$
 $(8, 1)], [(3, 1), (8, 1)]]),$
('H', $[[(-11, 11)], [(-7, -12), (-7, 9)], [(7, -12), (7, 9)],$
 $[(-7, -2), (7, -2)]]),$
('I', $[[(-4, 4)], [(0, -12), (0, 9)]]),$
('J', $[[(-8, 8)], [(4, -12), (4, 4), (3, 7), (2, 8), (0, 9), (-2, 9),$
 $(-4, 8), (-5, 7), (-6, 4), (-6, 2)]]),$
('K', $[[(-11, 10)], [(-7, -12), (-7, 9)], [(7, -12), (-7, 2)],$
 $[(-2, -3), (7, 9)]]),$
('L', $[[(-10, 7)], [(-6, -12), (-6, 9)], [(-6, 9), (6, 9)]]),$
('M', $[[(-12, 12)], [(-8, -12), (-8, 9)], [(-8, -12), (0, 9)],$
 $[(8, -12), (0, 9)], [(8, -12), (8, 9)]]),$
('N', $[[(-11, 11)], [(-7, -12), (-7, 9)], [(-7, -12), (7, 9)],$
 $[(7, -12), (7, 9)]]),$
('O', $[[(-11, 11)], [(-2, -12), (-4, -11), (-6, -9), (-7, -7),$
 $(-8, -4), (-8, 1), (-7, 4), (-6, 6), (-4, 8), (-2, 9), (2, 9),$
 $(4, 8), (6, 6), (7, 4), (8, 1), (8, -4), (7, -7), (6, -9),$
 $(4, -11), (2, -12), (-2, -12)]]),$
('P', $[[(-11, 10)], [(-7, -12), (-7, 9)], [(-7, -12), (2, -12),$
 $(5, -11), (6, -10), (7, -8), (7, -5), (6, -3), (5, -2), (2, -1),$
 $(-7, -1)]]),$
('Q', $[[(-11, 11)], [(-2, -12), (-4, -11), (-6, -9), (-7, -7),$
 $(-8, -4), (-8, 1), (-7, 4), (-6, 6), (-4, 8), (-2, 9), (2, 9),$
 $(4, 8), (6, 6), (7, 4), (8, 1), (8, -4), (7, -7), (6, -9),$
 $(4, -11), (2, -12), (-2, -12)], [(1, 5), (7, 11)]]),$
('R', $[[(-11, 10)], [(-7, -12), (-7, 9)], [(-7, -12), (2, -12),$
 $(5, -11), (6, -10), (7, -8), (7, -6), (6, -4), (5, -3), (2, -2),$
 $(-7, -2)], [(0, -2), (7, 9)]]),$

('S', [[(-10,10)], [(7,-9), (5,-11), (2,-12), (-2,-12),
 (-5,-11), (-7,-9), (-7,-7), (-6,-5), (-5,-4), (-3,-3),
 (3,-1), (5,0), (6,1), (7,3), (7,6), (5,8), (2,9), (-2,9),
 (-5,8), (-7,6)]]),
 ('T', [[(-8,8)], [(0,-12), (0,9)], [(-7,-12), (7,-12)]]),
 ('U', [[(-11,11)], [(-7,-12), (-7,3), (-6,6), (-4,8), (-1,9),
 (1,9), (4,8), (6,6), (7,3), (7,-12)]]),
 ('V', [[(-9,9)], [(-8,-12), (0,9)], [(8,-12), (0,9)]]),
 ('W', [[(-12,12)], [(-10,-12), (-5,9)], [(0,-12), (-5,9)],
 [(0,-12), (5,9)], [(10,-12), (5,9)]]),
 ('X', [[(-10,10)], [(-7,-12), (7,9)], [(7,-12), (-7,9)]]),
 ('Y', [[(-9,9)], [(-8,-12), (0,-2), (0,9)], [(8,-12), (0,-2)]]),
 ('Z', [[(-10,10)], [(7,-12), (-7,9)], [(-7,-12), (7,-12)],
 [(-7,9), (7,9)]]),
 ('a', [[(-9,10)], [(6,-5), (6,9)], [(6,-2), (4,-4), (2,-5),
 (-1,-5), (-3,-4), (-5,-2), (-6,1), (-6,3), (-5,6),
 (-3,8), (-1,9), (2,9), (4,8), (6,6)]]),
 ('b', [[(-10,9)], [(-6,-12), (-6,9)], [(-6,-2), (-4,-4),
 (-2,-5), (1,-5), (3,-4), (5,-2), (6,1), (6,3), (5,6),
 (3,8), (1,9), (-2,9), (-4,8), (-6,6)]]),
 ('c', [[(-9,9)], [(6,-2), (4,-4), (2,-5), (-1,-5), (-3,-4),
 (-5,-2), (-6,1), (-6,3), (-5,6), (-3,8), (-1,9), (2,9),
 (4,8), (6,6)]]),
 ('d', [[(-9,10)], [(6,-12), (6,9)], [(6,-2), (4,-4), (2,-5),
 (-1,-5), (-3,-4), (-5,-2), (-6,1), (-6,3), (-5,6),
 (-3,8), (-1,9), (2,9), (4,8), (6,6)]]),
 ('e', [[(-9,9)], [(-6,1), (6,1), (6,-1), (5,-3), (4,-4),
 (2,-5), (-1,-5), (-3,-4), (-5,-2), (-6,1), (-6,3),
 (-5,6), (-3,8), (-1,9), (2,9), (4,8), (6,6)]]),
 ('f', [[(-5,7)], [(5,-12), (3,-12), (1,-11), (0,-8), (0,9)],
 [(-3,-5), (4,-5)]]),
 ('g', [[(-9,10)], [(6,-5), (6,11), (5,14), (4,15), (2,16),
 (-1,16), (-3,15)], [(6,-2), (4,-4), (2,-5), (-1,-5),
 (-3,-4), (-5,-2), (-6,1), (-6,3), (-5,6), (-3,8), (-1,9),
 (2,9), (4,8), (6,6)]]),
 ('h', [[(-9,10)], [(-5,-12), (-5,9)], [(-5,-1), (-2,-4),
 (0,-5), (3,-5), (5,-4), (6,-1), (6,9)]]),
 ('i', [[(-4,4)], [(-1,-12), (0,-11), (1,-12), (0,-13), (-1,-12)],
 [(0,-5), (0,9)]]),
 ('j', [[(-5,5)], [(0,-12), (1,-11), (2,-12), (1,-13), (0,-12)],
 [(1,-5), (1,12), (0,15), (-2,16), (-4,16)]]),
 ('k', [[(-9,8)], [(-5,-12), (-5,9)], [(5,-5), (-5,5)], [(-1,1),
 (6,9)]]),

('l', [[(-4,4)], [(0,-12), (0,9)]]),
 ('m', [[(-15,15)], [(-11,-5), (-11,9)], [(-11,-1), (-8,-4),
 (-6,-5), (-3,-5), (-1,-4), (0,-1), (0,9)], [(0,-1),
 (3,-4), (5,-5), (8,-5), (10,-4), (11,-1), (11,9)]]),
 ('n', [[(-9,10)], [(-5,-5), (-5,9)], [(-5,-1), (-2,-4),
 (0,-5), (3,-5), (5,-4), (6,-1), (6,9)]]),
 ('o', [[(-9,10)], [(-1,-5), (-3,-4), (-5,-2), (-6,1), (-6,3),
 (-5,6), (-3,8), (-1,9), (2,9), (4,8), (6,6), (7,3),
 (7,1), (6,-2), (4,-4), (2,-5), (-1,-5)]]),
 ('p', [[(-10,9)], [(-6,-5), (-6,16)], [(-6,-2), (-4,-4),
 (-2,-5), (1,-5), (3,-4), (5,-2), (6,1), (6,3), (5,6),
 (3,8), (1,9), (-2,9), (-4,8), (-6,6)]]),
 ('q', [[(-9,10)], [(6,-5), (6,16)], [(6,-2), (4,-4), (2,-5),
 (-1,-5), (-3,-4), (-5,-2), (-6,1), (-6,3), (-5,6),
 (-3,8), (-1,9), (2,9), (4,8), (6,6)]]),
 ('r', [[(-7,6)], [(-3,-5), (-3,9)], [(-3,1), (-2,-2), (0,-4),
 (2,-5), (5,-5)]]),
 ('s', [[(-8,9)], [(6,-2), (5,-4), (2,-5), (-1,-5), (-4,-4),
 (-5,-2), (-4,0), (-2,1), (3,2), (5,3), (6,5), (6,6),
 (5,8), (2,9), (-1,9), (-4,8), (-5,6)]]),
 ('t', [[(-5,7)], [(0,-12), (0,5), (1,8), (3,9), (5,9)], [(-3,-5),
 (4,-5)]]),
 ('u', [[(-9,10)], [(-5,-5), (-5,5), (-4,8), (-2,9), (1,9),
 (3,8), (6,5)], [(6,-5), (6,9)]]),
 ('v', [[(-8,8)], [(-6,-5), (0,9)], [(6,-5), (0,9)]]),
 ('w', [[(-11,11)], [(-8,-5), (-4,9)], [(0,-5), (-4,9)],
 [(0,-5), (4,9)], [(8,-5), (4,9)]]),
 ('x', [[(-8,9)], [(-5,-5), (6,9)], [(6,-5), (-5,9)]]),
 ('y', [[(-8,8)], [(-6,-5), (0,9)], [(6,-5), (0,9), (-2,13),
 (-4,15), (-6,16), (-7,16)]]),
 ('z', [[(-8,9)], [(6,-5), (-5,9)], [(-5,-5), (6,-5)], [(-5,9),
 (6,9)]]),
 ('0', [[(-10,10)], [(-1,-12), (-4,-11), (-6,-8), (-7,-3),
 (-7,0), (-6,5), (-4,8), (-1,9), (1,9), (4,8), (6,5),
 (7,0), (7,-3), (6,-8), (4,-11), (1,-12), (-1,-12)]]),
 ('1', [[(-10,10)], [(-4,-8), (-2,-9), (1,-12), (1,9)]]),
 ('2', [[(-10,10)], [(-6,-7), (-6,-8), (-5,-10), (-4,-11),
 (-2,-12), (2,-12), (4,-11), (5,-10), (6,-8), (6,-6),
 (5,-4), (3,-1), (-7,9), (7,9)]]),
 ('3', [[(-10,10)], [(-5,-12), (6,-12), (0,-4), (3,-4), (5,-3),
 (6,-2), (7,1), (7,3), (6,6), (4,8), (1,9), (-2,9), (-5,8),
 (-6,7), (-7,5)]]),
 ('4', [[(-10,10)], [(3,-12), (-7,2), (8,2)], [(3,-12), (3,9)]]),

```

('5', [[(-10,10)], [(5,-12), (-5,-12), (-6,-3), (-5,-4),
  (-2,-5), (1,-5), (4,-4), (6,-2), (7,1), (7,3), (6,6),
  (4,8), (1,9), (-2,9), (-5,8), (-6,7), (-7,5)]]),
('6', [[(-10,10)], [(6,-9), (5,-11), (2,-12), (0,-12), (-3,-11),
  (-5,-8), (-6,-3), (-6,2), (-5,6), (-3,8), (0,9), (1,9),
  (4,8), (6,6), (7,3), (7,2), (6,-1), (4,-3), (1,-4),
  (0,-4), (-3,-3), (-5,-1), (-6,2)]]),
('7', [[(-10,10)], [(7,-12), (-3,9)], [(-7,-12), (7,-12)]]),
('8', [[(-10,10)], [(-2,-12), (-5,-11), (-6,-9), (-6,-7),
  (-5,-5), (-3,-4), (1,-3), (4,-2), (6,0), (7,2), (7,5),
  (6,7), (5,8), (2,9), (-2,9), (-5,8), (-6,7), (-7,5),
  (-7,2), (-6,0), (-4,-2), (-1,-3), (3,-4), (5,-5), (6,-7),
  (6,-9), (5,-11), (2,-12), (-2,-12)]]),
('9', [[(-10,10)], [(6,-5), (5,-2), (3,0), (0,1), (-1,1),
  (-4,0), (-6,-2), (-7,-5), (-7,-6), (-6,-9), (-4,-11),
  (-1,-12), (0,-12), (3,-11), (5,-9), (6,-5), (6,0),
  (5,5), (3,8), (0,9), (-2,9), (-5,8), (-6,6)]]),
(' ', [[(-8,8)], []]),
('-', [[(-13,13)], [(-9,0), (9,0)]]),
('.', [[(-5,5)], [(0,7), (-1,8), (0,9), (1,8), (0,7)]]),
(',', [[(-5,5)], [(1,8), (0,9), (-1,8), (0,7), (1,8), (1,10),
  (0,12), (-1,13)]]),
(':', [[(-5,5)], [(0,-5), (-1,-4), (0,-3), (1,-4), (0,-5)],
  [(0,7), (-1,8), (0,9), (1,8), (0,7)]]),
('; ', [[(-5,5)], [(0,-5), (-1,-4), (0,-3), (1,-4), (0,-5)],
  [(1,8), (0,9), (-1,8), (0,7), (1,8), (1,10), (0,12),
  (-1,13)]]),
('!', [[(-5,5)], [(0,-12), (0,2)], [(0,7), (-1,8), (0,9),
  (1,8), (0,7)]]),
('?', [[(-9,9)], [(-6,-7), (-6,-8), (-5,-10), (-4,-11),
  (-2,-12), (2,-12), (4,-11), (5,-10), (6,-8), (6,-6),
  (5,-4), (4,-3), (0,-1), (0,2)], [(0,7), (-1,8), (0,9),
  (1,8), (0,7)]]
]

```

Voimme nyt käyttää kirjastossa lueteltuja kärkipistekoordinaatteja Hersheyn kirjainten piirtämiseen pääohjelmassa.

```

import Hershey
import Data.Char (toUpper,ord,chr)
import Data.List.Split (splitPlaces)
import qualified Data.Map as Map
import Control.Concurrent.MVar
import System.IO.Unsafe

```



```

import Graphics.UI.Gtk
import qualified Graphics.Rendering.Cairo as C
import qualified Graphics.UI.Gtk.Gdk.EventM as M
import System.Glib.UTFString (glibToString)

intToDouble :: Int -> Double
intToDouble = fromRational . toRational

int = fromIntegral

takeWhileSumLt mx sm (x:xs)
  | sm == 0      = x : takeWhileSumLt mx (x + sp) xs
  | sm + x <= mx = x : takeWhileSumLt mx (sm + x + sp) xs
  | otherwise    = []
  where
    sp = space
takeWhileSumLt _ _ _ = []

letterXs sm (x:xs) =
  (sm, sm + x1) : letterXs (sm + x1) xs      -- (start,end)
  where
    x1 = tellLen x
letterXs _ _ = []

fstS xs = [a | (a,b) <- xs]
sndS xs = [b | (a,b) <- xs]

wordXs sm (xs:xss) =
  (xs1,xs2) : wordXs newX xss
  where
    newX = xs2 + space
    xs1 = minimum (fstS letters)
    xs2 = maximum (sndS letters)
    letters = letterXs sm xs
wordXs _ _ = []

lineXys y (xss:xsss) =
  xs1 : lineXys (y+lineSpace) xsss
  where
    xs1 = [(x,y) | x <- fstS (wordXs 0 xss)]
lineXys _ _ = []

head1 (x:xs) = x

```

```

head1 _ = []

head2 (x:xs) = x
head2 _ = (0,0)

tail2 (x:xs) = xs
tgal2 _ = []

lrHands c = (head2 . head1) (verts c)
lrHands1 ch = (head2 . head1) ch

tellLen c = abs a + abs b
  where
    (a,b) = lrHands c

cutList maxLen [] = [[]]
cutList maxLen xs =
  fstLn : cutList maxLen (drop (length fstLn) xs)
  where
    fstLn = takeWhileSumLt maxLen 0 xs

h = hershey2
map1 = Map.fromList h
verts c = map1 Map.! c

skale = 3
xMargin = 50
yMargin = 33 + 41
lineSpace = 21 + 13
space = 16
(imgW,imgH) = (854,480)

origRight hands (x,y) = (x + (-leftHandPos), y)
  where
    (leftHandPos,rightHandPos) = hands

move0 (x1,y1) (x2,y2) = (x1+x2,y1+y2)
move2 (x1,y1) pts = [(x1+x2,y1+y2)| (x2,y2) <- pts]
scale0 k (x1,y1) = (k*x1,k*y1)

mkTxt1 map1 chars = h4
  where
    -- [[('L',[(407,240),(407,303)],[(407,303),

```

```

-- (443,303)]))]]
h4 = [[(c,[map sm1 pls | pls <- ch]) | (c,ch) <- ws]
      | ws <- h3]
sm1 (x,y) =
  (skale * x + xMargin, skale * y + yMargin)
-- ('M',[[ (61,56),(61,77)],[(61,56),(69,77)],
-- [(77,56),(69,77)],[(77,56),(77,77)]]])
h3 = [[(c, mv2 (mv1 (x,y) ch) ch)
      | (c,(x,y),ch) <- ws] | ws <- h2]
mv1 (x,y) ch = origRight (lrHands1 ch) (x,y)
mv2 (x,y) ch = map (move2 (x,y)) (tail ch)
-- [[('L',(115,68),[(-10,7)],[(-6,-12),(-6,9)],
-- [(-6,9),(6,9)])]]])
h2 = [[(c,(x+x1,y),verts c)
      | (c,x1) <- zip cx (fst$ letterXs 0 cx))]
      | ((x,y),cx) <- h1]
-- [( (0,0),"BON"),((81,0),"DIE!") ]
h1 = zip (concat lx) brokead
lx = lineXys 0 cd
cd = splitPlaces [length x | x <- d1] cxx
d1 = cutList maxLen ls1
maxLen = (imgW - (2*xMargin)) `div` skale
ls1 = [sum [tellLen c | c <- cx] | cx <- cxx]
cxx = [[c | c <- bx] | bx <- brokead]
brokead = words chars

mtxt = mkTxt1 map1 text
where
  text = -- map toUpper
    "Pariisista Helsinkiin on 1907 km."
    --"Helsingistä Pariisiin on 1907 kilometriä."
    --"Bon die! Hic es Alec e Adam. Illes es amicos. "

main = do
  initGUI
  var <- newMVar (1.0,0.0,0.0)
  vPos <- newMVar (None,0.0,0.0)
  window <- windowNew
  canvas <- drawingAreaNew
  surf <- return $ unsafeLoadPNG "tausta-854x480px.png"
  widgetAddEvents canvas [Button1MotionMask]
  widgetSetSizeRequest canvas (int imgW) (int imgH)
  centerImg var surf canvas

```

```

canvas `on` motionNotifyEvent $ do
  (mouseX,mouseY) <- eventCoordinates
  t <- M.eventTime
  C.liftIO $ changePos vPos var surf canvas mouseX mouseY
  C.liftIO $ logMsg 0 ("Motion Time: " ++ s t)
  return False
window `on` keyPressEvent $ tryEvent $ do
  key <- eventKeyName
  keyInput var surf canvas (glibToString key)
  C.liftIO $ updateCanvas1 var canvas surf
  return ()
canvas `on` buttonPressEvent $ tryEvent $ do
  (mouseX,mouseY) <- printMouse
  C.liftIO $ printPointer canvas
  C.liftIO $ printMVar var mouseX mouseY
  C.liftIO $ modifyMVar_ vPos (\_ ->
    return (Press,mouseX,mouseY))
canvas `on` buttonReleaseEvent $ tryEvent $ do
  (mouseX,mouseY) <- M.eventCoordinates
  m <- M.eventModifier
  b <- M.eventButton
  (cause,vPosX,vPosY) <- C.liftIO $ readMVar vPos
  C.liftIO $ release cause b var vPosX vPosY
canvas `on` scrollEvent $ tryEvent $ do
  (mouseX,mouseY) <- M.eventCoordinates
  m <- M.eventModifier
  d <- M.eventScrollDirection
  t <- M.eventTime
  C.liftIO $ changeRef var d mouseX mouseY
  C.liftIO $ updateCanvas1 var canvas surf
  C.liftIO $ logMsg 0 ("Scroll: " ++ s t ++ s mouseX ++
    s mouseY ++ s m ++ s d)
onDestroy window mainQuit
onExpose canvas $ const (updateCanvas1 var canvas surf)
set window [containerChild := canvas]
widgetShowAll window
mainGUI

```

```

data EvtType = Press | Release | Move | Scroll | None

```

```

release Press button var mouseX mouseY = do
  (varS,varX,varY) <- readMVar var
  let

```

```

    x = (mouseX - varX) / varS
    y = (mouseY - varY) / varS
C.liftIO $ logMsg 0
  ("Add point: " ++ s x ++ s y ++ s button)
C.liftIO $ logMsg 1 (s x ++ s y)

release _ button var x y = do
  C.liftIO $ logMsg 0 ("Ignore: " ++ s x ++ s y)

changePos vPos var surf canvas mouseX mouseY = do
  (cause,vPosX,vPosY) <- readMVar vPos
  (scaleOld,oldX,oldY) <- readMVar var
  let
    dx = vPosX - mouseX
    dy = vPosY - mouseY
  modifyMVar_ var (\_ ->
    return (scaleOld,oldX-dx,oldY-dy))
  modifyMVar_ vPos (\_ -> return (Move,mouseX,mouseY))
  updateCanvas1 var canvas surf

s x = show x ++ " "

printMouse = do
  (mouseX,mouseY) <- M.eventCoordinates
  C.liftIO $ logMsg 0 ("Mouse: " ++ s mouseX ++ s mouseY)
  return (mouseX,mouseY)

printPointer canvas = do
  (widX,widY) <- widgetGetPointer canvas
  logMsg 0 ("Widget: " ++ s widX ++ s widY)

printMVar var mouseX mouseY = do
  (varS,varX,varY) <- readMVar var
  let
    x = (mouseX - varX) / varS
    y = (mouseY - varY) / varS
  logMsg 0 ("MVar: " ++ s varS ++ s varX ++ s varY)
  logMsg 0 ("Calc: " ++ s x ++ s y)

centerImg var surf canvas = do
  w1 <- C.imageSurfaceGetWidth surf
  h1 <- C.imageSurfaceGetHeight surf
  (w2,h2) <- widgetGetSizeRequest canvas

```

```

let
  dh = intToDouble (h2 - h1)
  dw = intToDouble (w2 - w1)
  modifyMVar_ var (\_ -> return (1.0,dw/2,dh/2))

strokePolyLine img pts = do
  C.renderWith img ( do
    C.setSourceRGB 0.7 0.7 0.7
    C.setLineWidth 3.2
    C.setLineCap C.LineCapRound
    (\(x1,y1) -> C.moveTo (d x1) (d y1)) (head2 pts)
    mapM_ (\(x,y) -> C.lineTo (d x) (d y)) (tail pts)
    C.stroke
  )
  where
    d = (intToDouble . fromIntegral)

strokeChar surf char = mapM_ (strokePolyLine surf) char

strokeWord surf txt =
  mapM_ (\(c,ch) -> strokeChar surf ch) txt

strokeText surf mtxt = mapM_ (strokeWord surf) mtxt
  -- ('M',[[ (61,56),(61,77)],[(61,56),(69,77)],
  -- [(77,56),(69,77)],[(77,56),(77,77)]]))

keyInput var surf canvas key = do
  C.liftIO $ logMsg 1 ("Key Input: " ++ key)
  case key of
    "q" -> do
      C.liftIO $ mainQuit
    "1" -> do
      C.liftIO $ centerImg var surf canvas
    "m" -> do
      C.liftIO $ putStrLn (show mtxt)
    "n" -> do
      C.liftIO $ putStrLn (show nodelist)
    "p" -> do
      C.liftIO $ strokeText surf mtxt
    "s" -> do
      C.liftIO $ createPNG surf "test-2.png"

changeRef var d mouseX mouseY = do

```

```

(scaleOld,oldX,oldY) <- readMVar var
let
  scaleD = scale1 d
  scaleNew = scaleD * scaleOld
  dx = (mouseX - oldX) * (scaleD - 1)
  dy = (mouseY - oldY) * (scaleD - 1)
  newX = oldX - dx
  newY = oldY - dy
  result = (scaleNew,newX,newY)
modifyMVar_ var (\_ -> return result)
logMsg 0 ("Change MVar: " ++ s scaleNew ++
  s newX ++ s newY)
where
  factor = 5/4
  scale1 ScrollUp    = factor
  scale1 ScrollDown = 1/factor

updateCanvas1 var canvas surf = do
  win <- widgetGetDrawWindow canvas
  (width, height) <- widgetGetSize canvas
  renderWithDrawable win $
    paintImage1 var surf
  return True

createPNG img fileName = do
  C.surfaceWriteToPNG img fileName
  C.liftIO ( do
    logMsg 1 "Save Image"
    logMsg 1 ("Created: " ++ fileName)
    logMsg 1 ("Size: " ++ show imgW ++ "x" ++ show imgH ++ " pixels")
  )

imageSurfaceCreateFromPNG :: FilePath -> IO C.Surface
imageSurfaceCreateFromPNG file =
  C.withImageSurfaceFromPNG file $ \png -> do
    C.liftIO $ logMsg 0 "Load Image"
    w <- C.renderWith png $ C.imageSurfaceGetWidth png
    h <- C.renderWith png $ C.imageSurfaceGetHeight png
    surf <- C.createImageSurface C.FormatRGB24 w h
    C.renderWith surf $ do
      C.setSourceSurface png 0 0
      C.paint
    return surf

```

```

unsafeLoadPNG file =
  unsafePerformIO $ imageSurfaceCreateFromPNG file

paintImage1 var surf = do
  (sc,x,y) <- C.liftIO $ readMVar var
  C.setSourceRGB 1 1 1
  C.paint
  C.translate x y
  C.scale sc sc
  C.liftIO $ logMsg 0
    ("Paint Image: " ++ s sc ++ s x ++ s y)
  C.setSourceSurface surf 0 0
  C.paint

logMsg 0 s = do
  return ()
logMsg 1 s = do
  putStrLn s
  return ()

```

Kuvassa 1 on ohjelman tallentama PNG-kuva.



Kuva 1. Esimerkkiohjelman tallentama PNG-muotoinen kokeilukuva.

Ä- ja Ö-kirjaimen lisääminen

Lisätäksemme kokoelmaan Ä- ja Ö-kirjaimet, kopioimme sopivan kokoisen pisteen i-kirjaimesta:

```
('i', [[(-4,4)], [(-1,-12), (0,-11), (1,-12), (0,-13), (-1,-12)],  
      [(0,-5), (0,9)]]),
```

Voimme siirtää sen haluamaamme paikkaan komennolla move0:

```
> map (move0 (-3,-3)) [(-1,-12), (0,-11), (1,-12), (0,-13), (-1,-12)]  
[(-4,-15), (-3,-14), (-2,-15), (-3,-16), (-4,-15)]  
> map (move0 (3,-3)) [(-1,-12), (0,-11), (1,-12), (0,-13), (-1,-12)]  
[(2,-15), (3,-14), (4,-15), (3,-16), (2,-15)]
```

Muodostamme Ä-kirjaimen A-kirjaimesta:

```
('A', [[(-9,9)], [(0,-12), (-8,9)], [(0,-12), (8,9)], [(-5,2),  
      (5,2)]])
```

```
('Ä', [[(-9,9)], [(0,-12), (-8,9)], [(0,-12), (8,9)], [(-5,2),  
      (5,2)],  
      [(-3,-15), (-2,-14), (-1,-15), (-2,-16), (-3,-15)],  
      [(1,-15), (2,-14), (3,-15), (2,-16), (1,-15)]])
```

Ö-kirjaimen O-kirjaimesta:

```
('O', [[(-11,11)], [(-2,-12), (-4,-11), (-6,-9), (-7,-7),  
      (-8,-4), (-8,1), (-7,4), (-6,6), (-4,8), (-2,9), (2,9),  
      (4,8), (6,6), (7,4), (8,1), (8,-4), (7,-7), (6,-9),  
      (4,-11), (2,-12), (-2,-12)]])
```

```
('Ö', [[(-11,11)], [(-2,-12), (-4,-11), (-6,-9), (-7,-7),  
      (-8,-4), (-8,1), (-7,4), (-6,6), (-4,8), (-2,9), (2,9),  
      (4,8), (6,6), (7,4), (8,1), (8,-4), (7,-7), (6,-9),  
      (4,-11), (2,-12), (-2,-12)],  
      [(-4,-15), (-3,-14), (-2,-15), (-3,-16), (-4,-15)],  
      [(2,-15), (3,-14), (4,-15), (3,-16), (2,-15)]])
```

Pikku-ä syntyy a-kirjaimesta:

```
('a', [[(-9,10)], [(6,-5), (6,9)], [(6,-2), (4,-4), (2,-5),  
      (-1,-5), (-3,-4), (-5,-2), (-6,1), (-6,3), (-5,6),  
      (-3,8), (-1,9), (2,9), (4,8), (6,6)]])
```

```
('ä', [[(-9,10)], [(6,-5), (6,9)], [(6,-2), (4,-4), (2,-5),  
      (-1,-5), (-3,-4), (-5,-2), (-6,1), (-6,3), (-5,6),  
      (-3,8), (-1,9), (2,9), (4,8), (6,6)],
```

$[(-3, -10), (-2, -9), (-1, -10), (-2, -11), (-3, -10)],$
 $[(2, -10), (3, -9), (4, -10), (3, -11), (2, -10)]]]$

ja lopuksi pikku-ö o-kirjaimesta:

$('o', [[(-9, 10)], [(-1, -5), (-3, -4), (-5, -2), (-6, 1), (-6, 3),$
 $(-5, 6), (-3, 8), (-1, 9), (2, 9), (4, 8), (6, 6), (7, 3),$
 $(7, 1), (6, -2), (4, -4), (2, -5), (-1, -5)]]])$

$('ö', [[(-9, 10)], [(-1, -5), (-3, -4), (-5, -2), (-6, 1), (-6, 3),$
 $(-5, 6), (-3, 8), (-1, 9), (2, 9), (4, 8), (6, 6), (7, 3),$
 $(7, 1), (6, -2), (4, -4), (2, -5), (-1, -5)],$
 $[(-3, -10), (-2, -9), (-1, -10), (-2, -11), (-3, -10)],$
 $[(2, -10), (3, -9), (4, -10), (3, -11), (2, -10)]]])$

Kuvassa 2 voimme nähdä miltä uudet kirjaimet näyttävät valmiina.



Kuva 2. Kirjaimet Ä, Ö, ä ja ö.

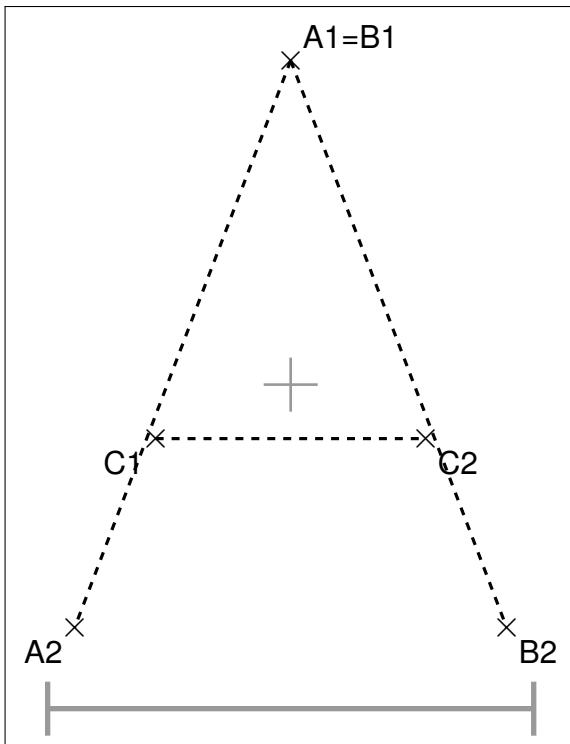
Kirjainten muodostuminen viivoista

Hersheyn kirjaimet muodostuvat kärkipistekoordinaattien muodostamista viivajonoista.

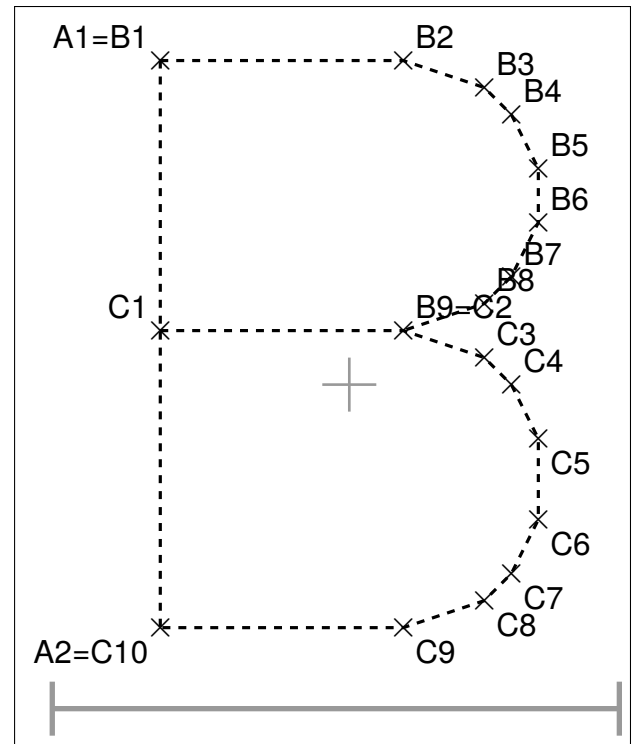
Seuraavilla sivuilla esittelemme solmujonoina suuraakkoset A–Z (koodit 89–114), numerot 0–9 (koodit 251–260), pisteen (261), pilkun (262), kaksoispisteen (263), puolipisteen (264), huutomerkin (265) ja kysymysmerkin (266) (kuvat 3–44).

Solmujonot saamme solmujonojen koordinaateista purkamalla teksti sanojen listaksi, sanojen lista kirjainhahmoiksi, kirjainhahmot viivajonoiksi ja lopuksi viivajonot kärkipistekoordinaateiksi.

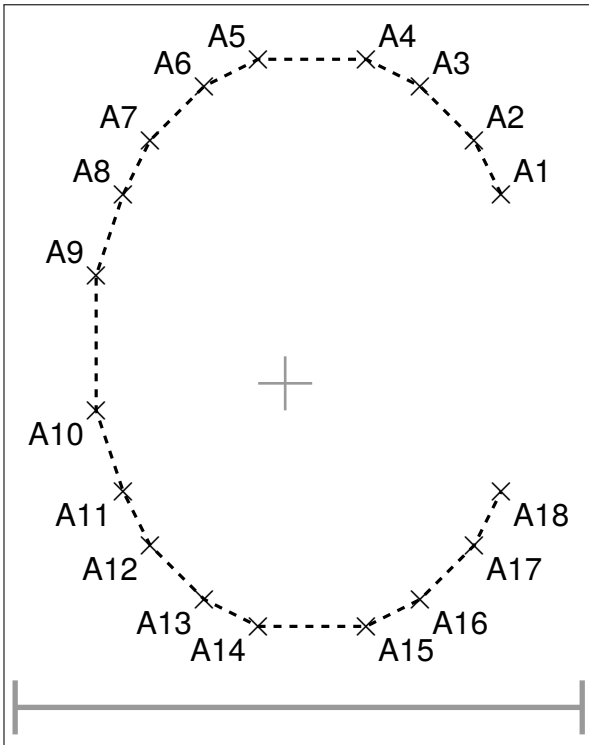
```
node1 h (i,pt) = (name,pt)
  where
    name = [chr (ord 'A' + (h-1))] ++ show i
node2 (h,pts) = map (node1 h) (zip [1..] pts)
node3 pls = map node2 (zip [1..] pls)
node4 ch = map node3 ch
--[['I',[[("A1",(188,344)),("A2",(188,407))]]]]
nodelist = lst
  where
    lst = [(c,node3 ch) | (c,ch) <- ws] | ws <- h4]
    h4 = mtxt
```



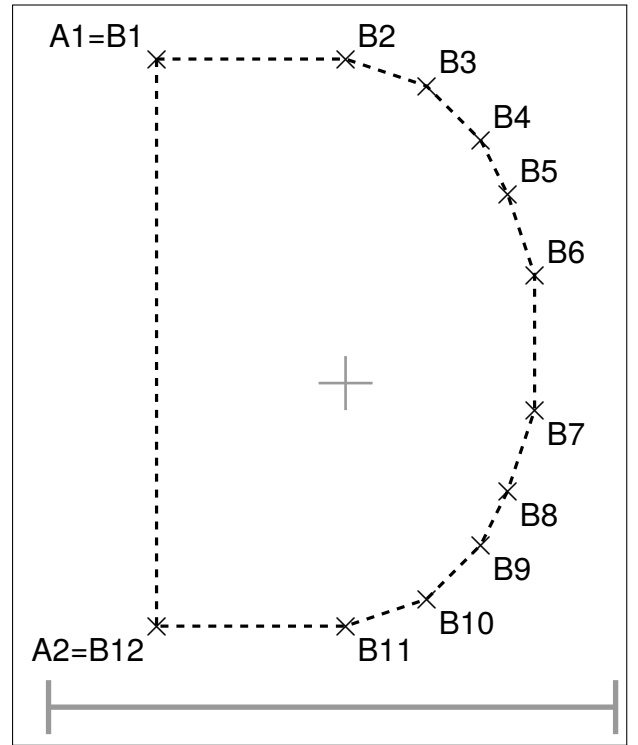
Kuva 3. A-kirjain.



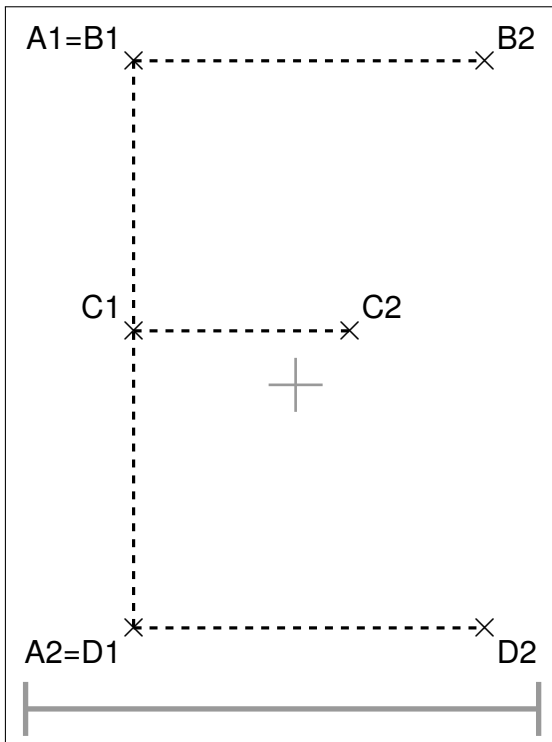
Kuva 4. B-kirjain.



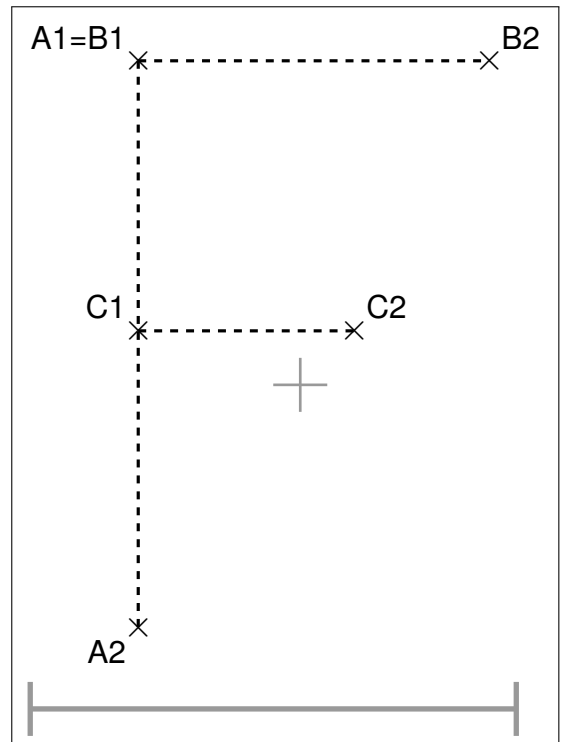
Kuva 5. C-kirjain.



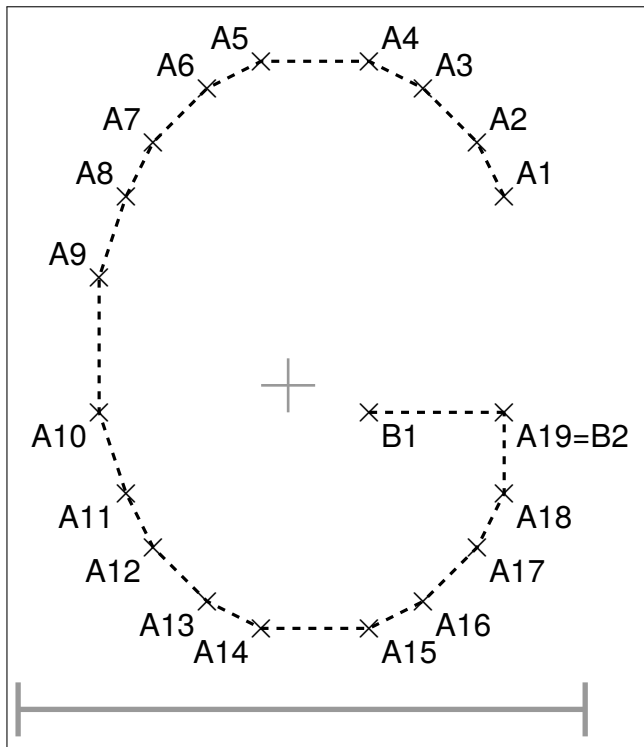
Kuva 6. D-kirjain.



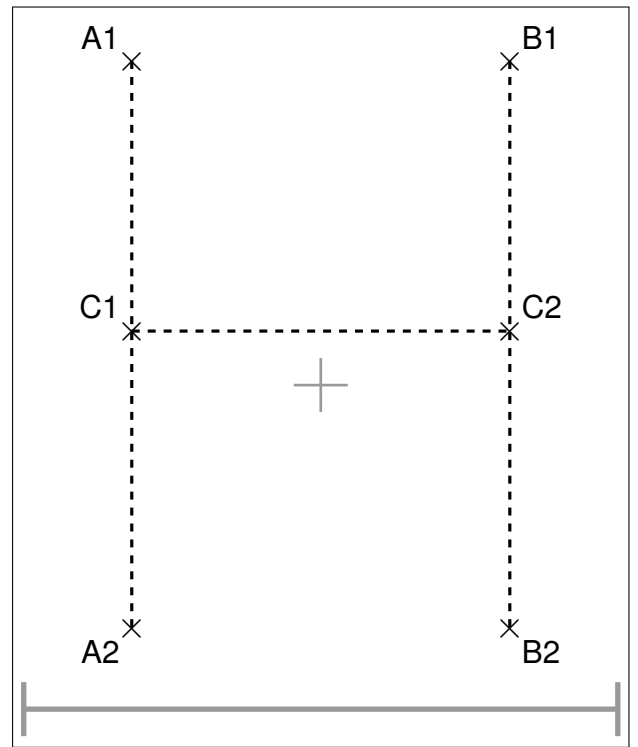
Kuva 7. E-kirjain.



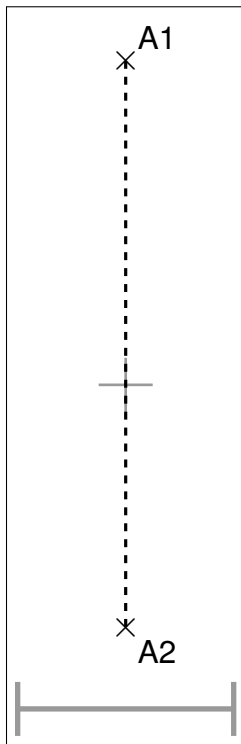
Kuva 8. F-kirjain.



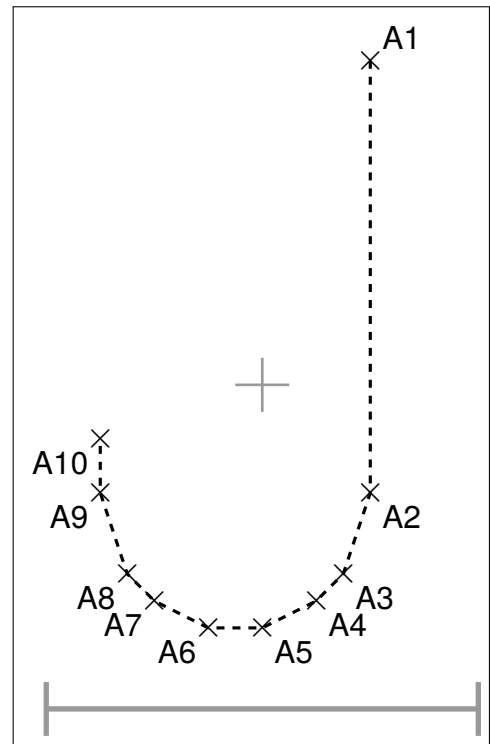
Kuva 9. G-kirjain.



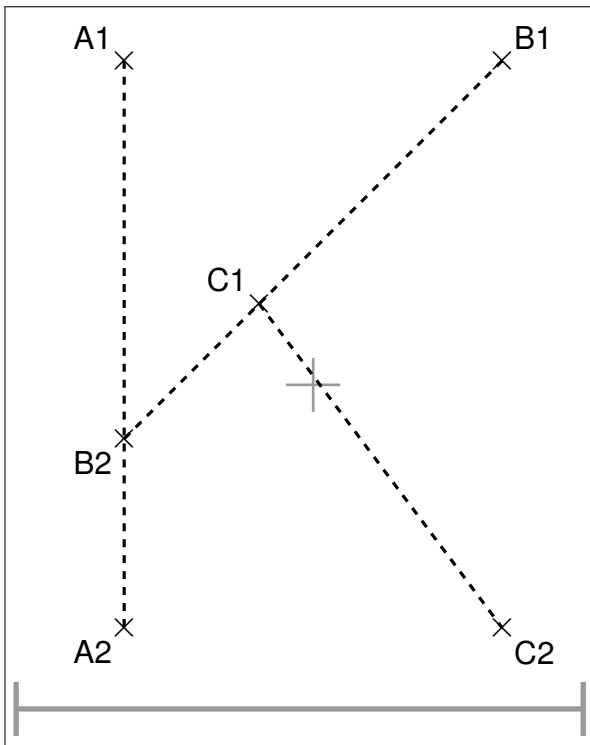
Kuva 10. H-kirjain.



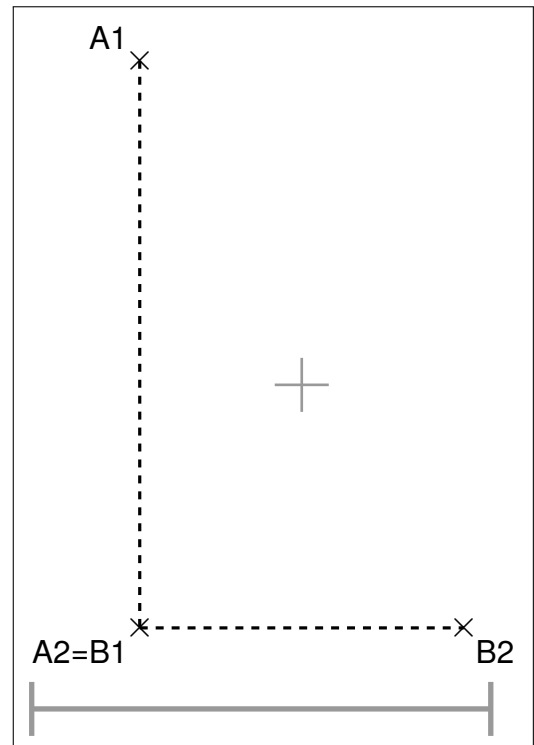
Kuva 11. I-kirjain.



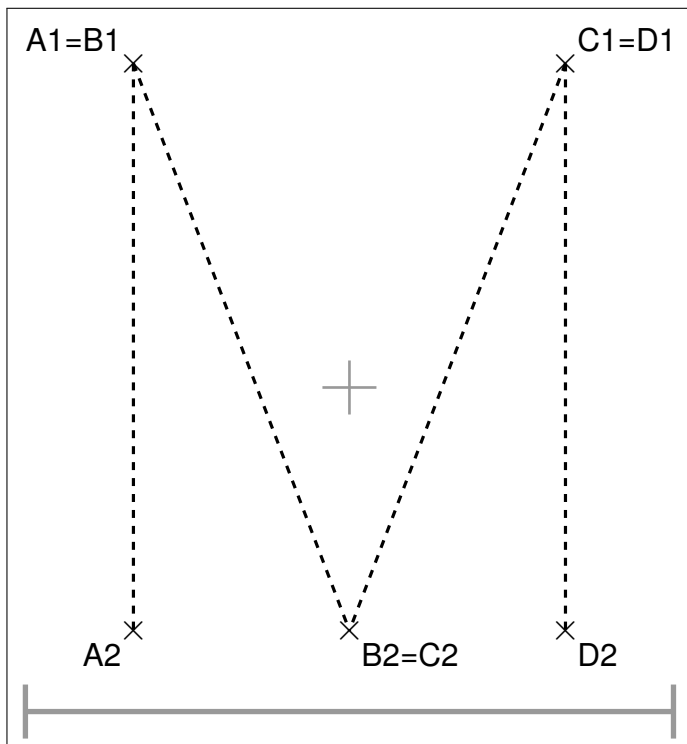
Kuva 12. J-kirjain.



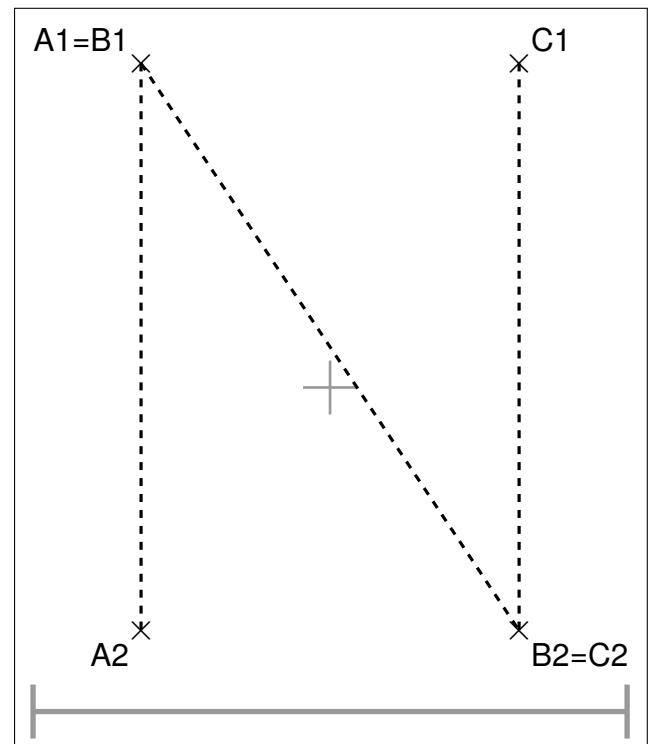
Kuva 13. K-kirjain.



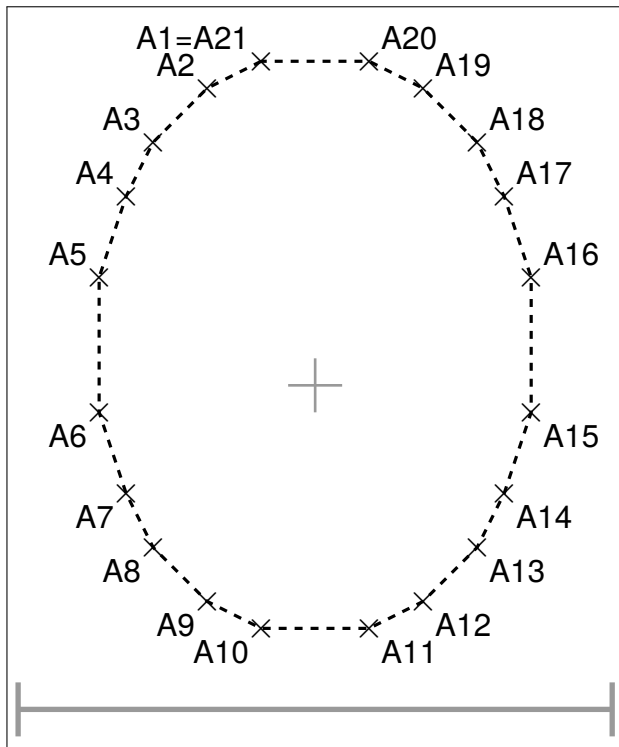
Kuva 14. L-kirjain.



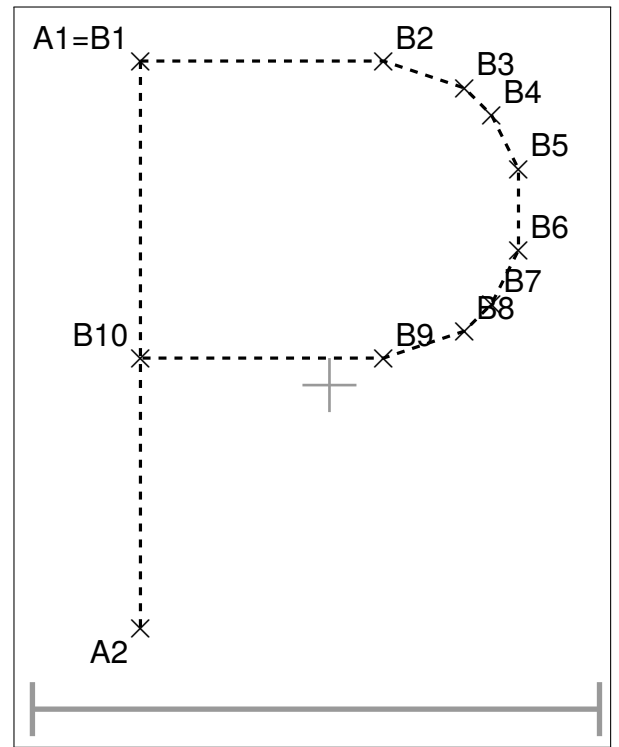
Kuva 15. M-kirjain.



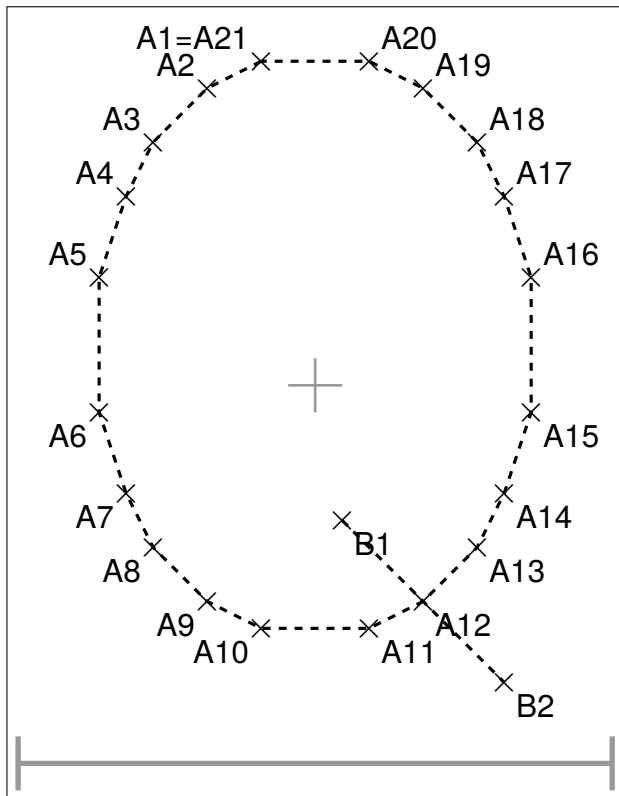
Kuva 16. N-kirjain.



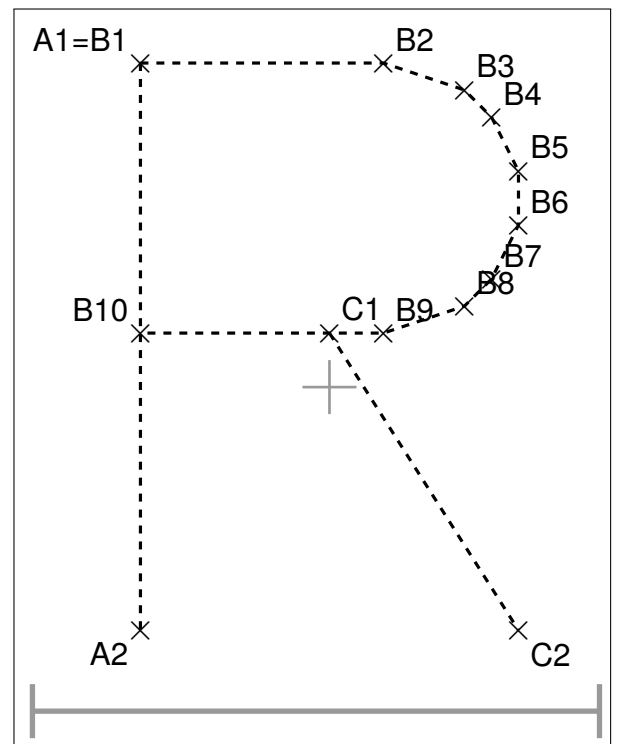
Kuva 17. O-kirjain.



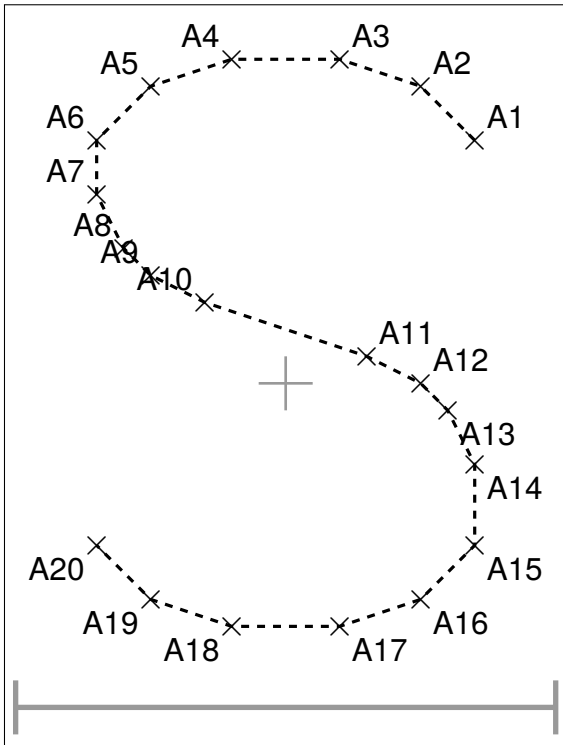
Kuva 18. P-kirjain.



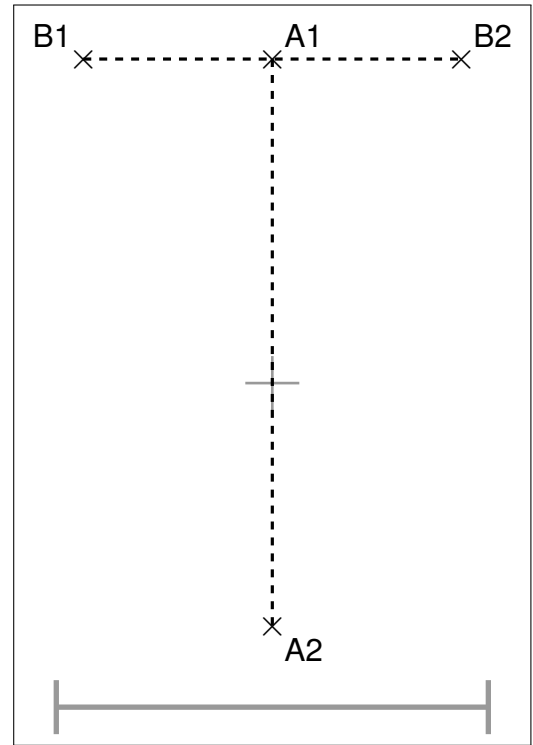
Kuva 19. Q-kirjain.



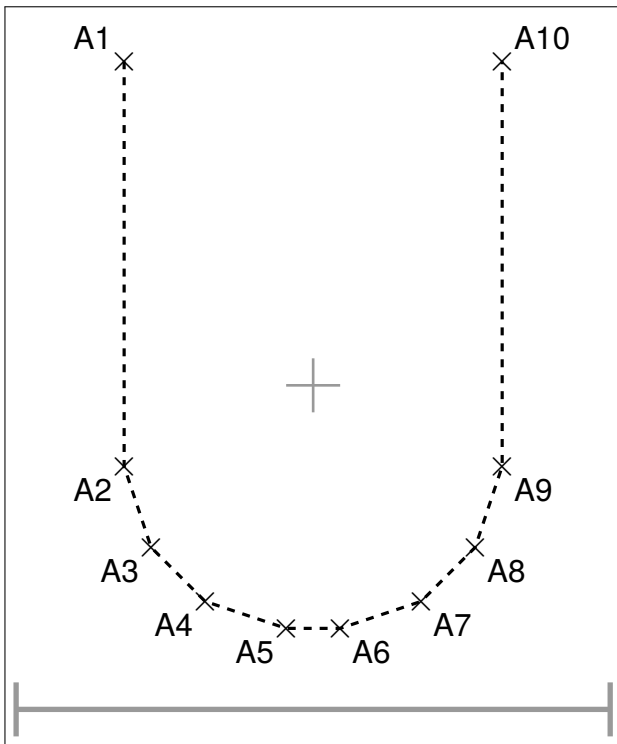
Kuva 20. R-kirjain.



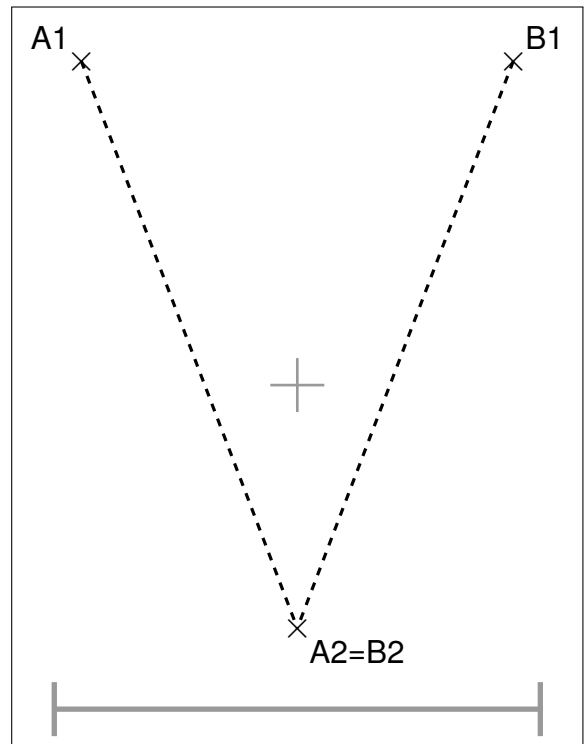
Kuva 21. S-kirjain.



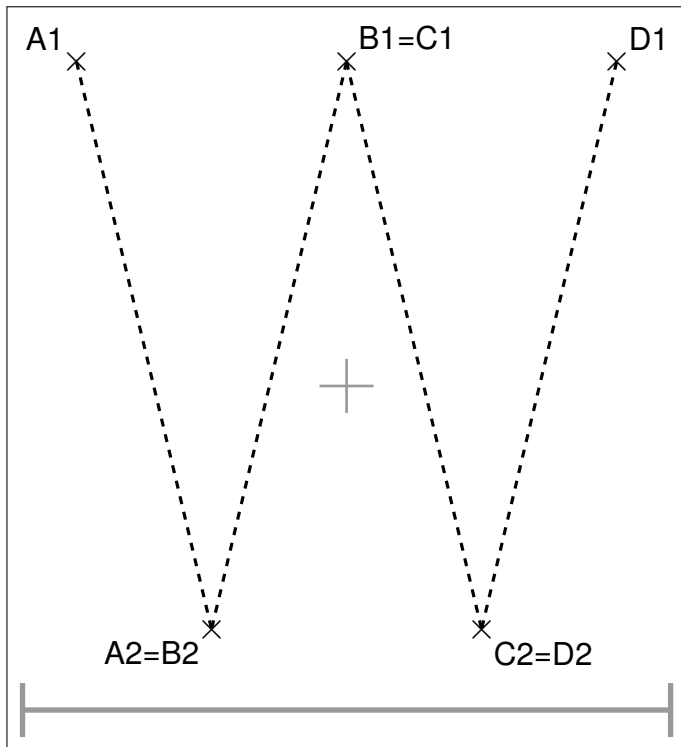
Kuva 22. T-kirjain.



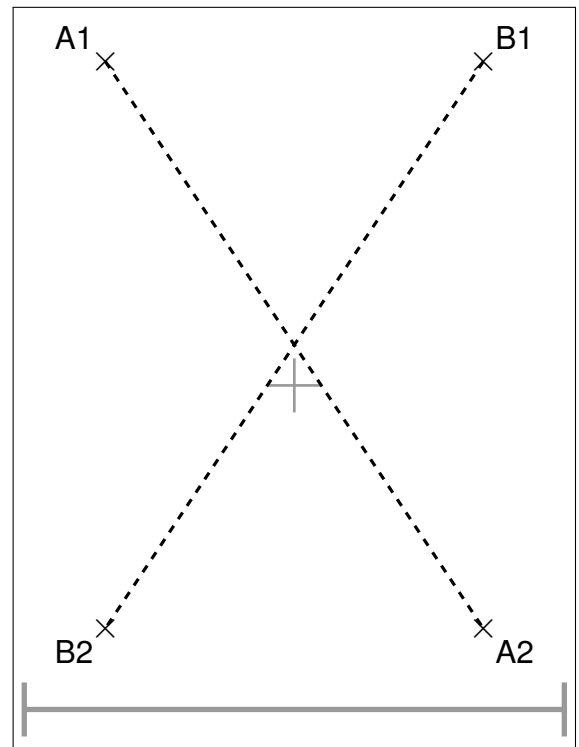
Kuva 23. U-kirjain.



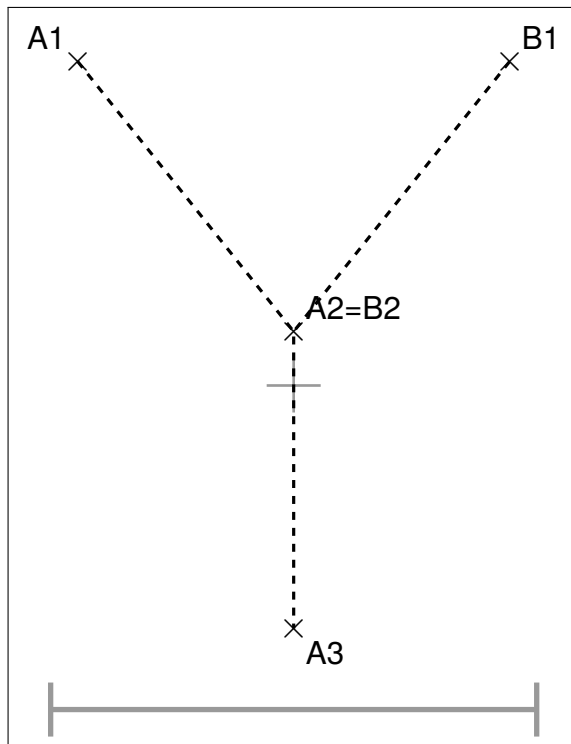
Kuva 24. V-kirjain.



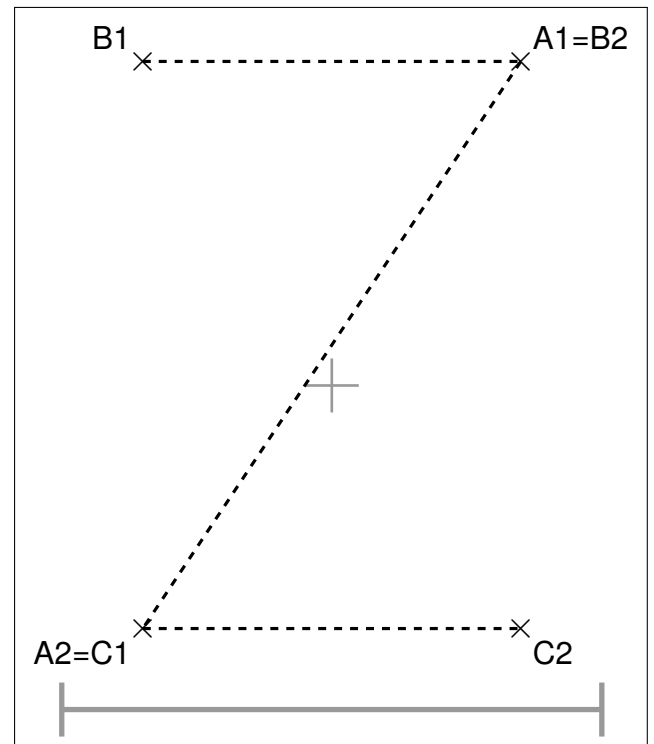
Kuva 25. W-kirjain.



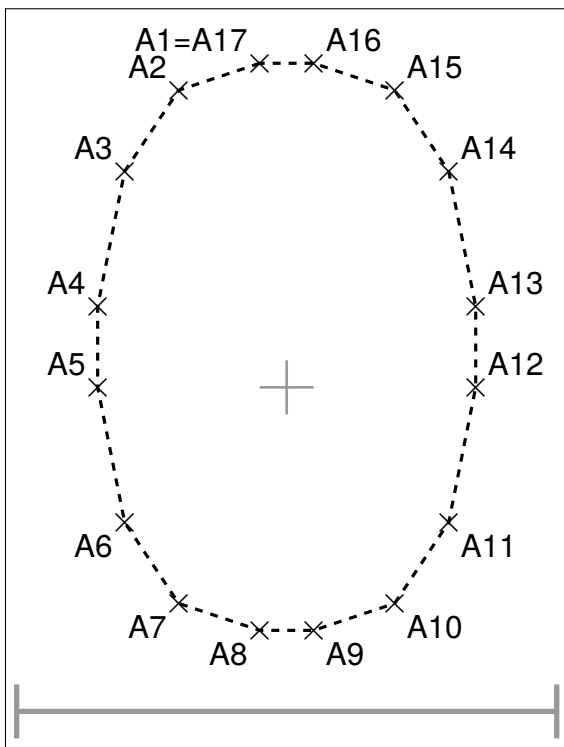
Kuva 26. X-kirjain.



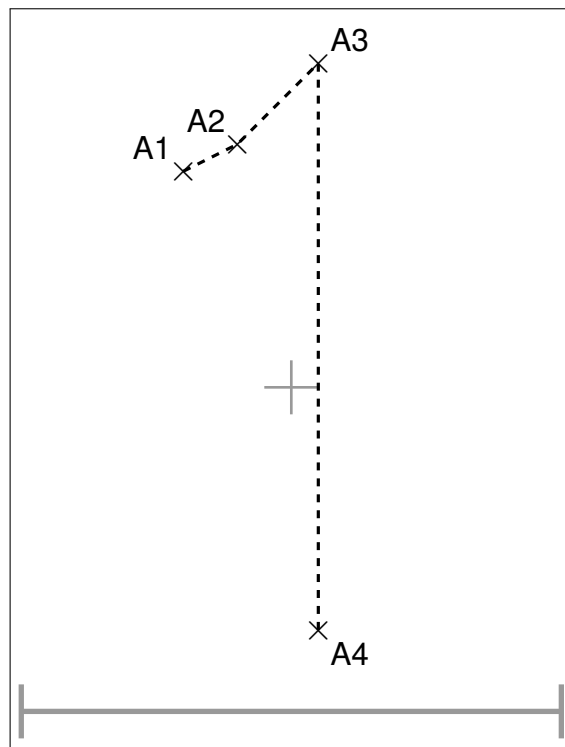
Kuva 27. Y-kirjain.



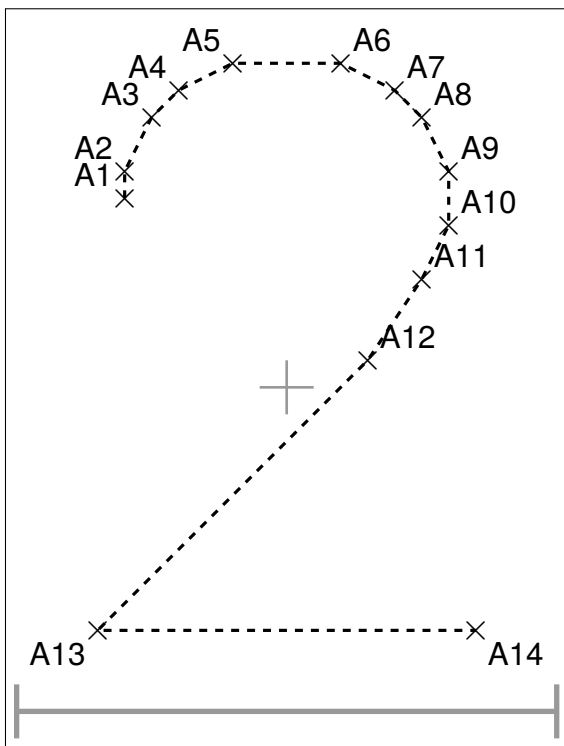
Kuva 28. Z-kirjain.



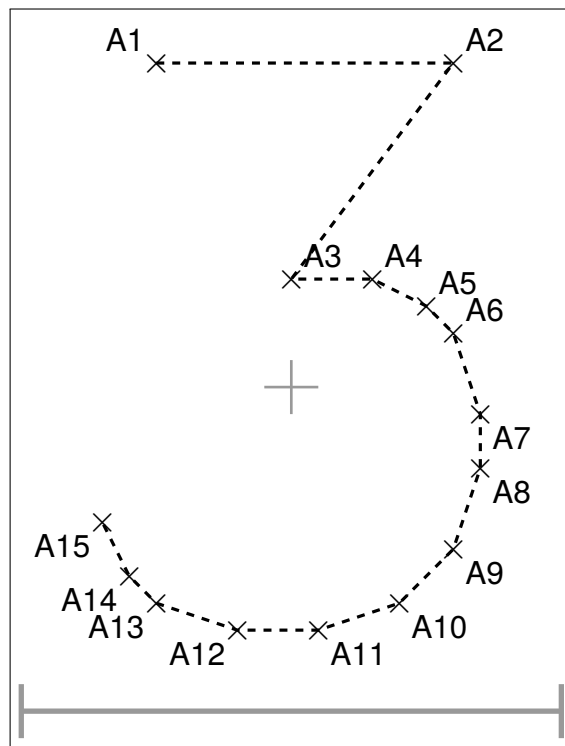
Kuva 29. Numero 0.



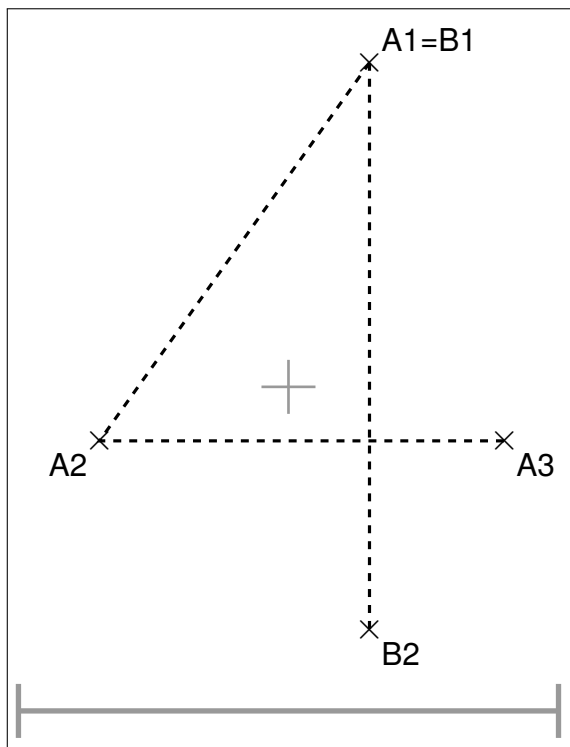
Kuva 30. Numero 1.



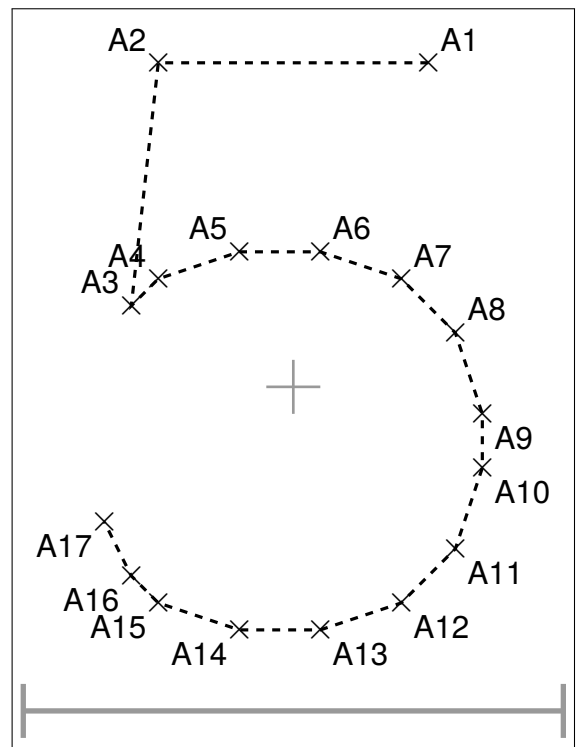
Kuva 31. Numero 2.



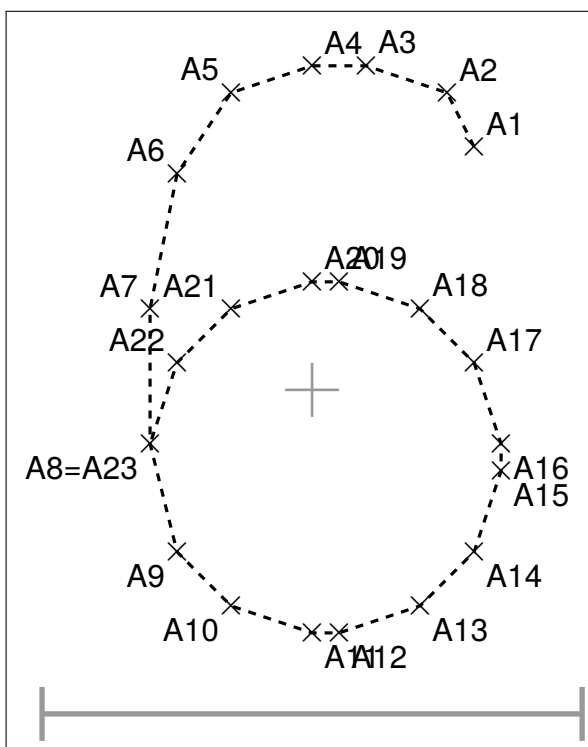
Kuva 32. Numero 3.



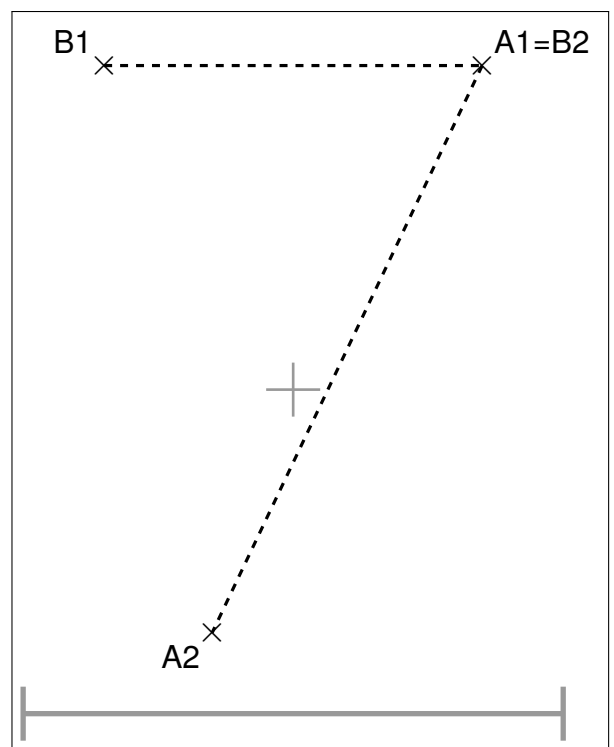
Kuva 33. Numero 4.



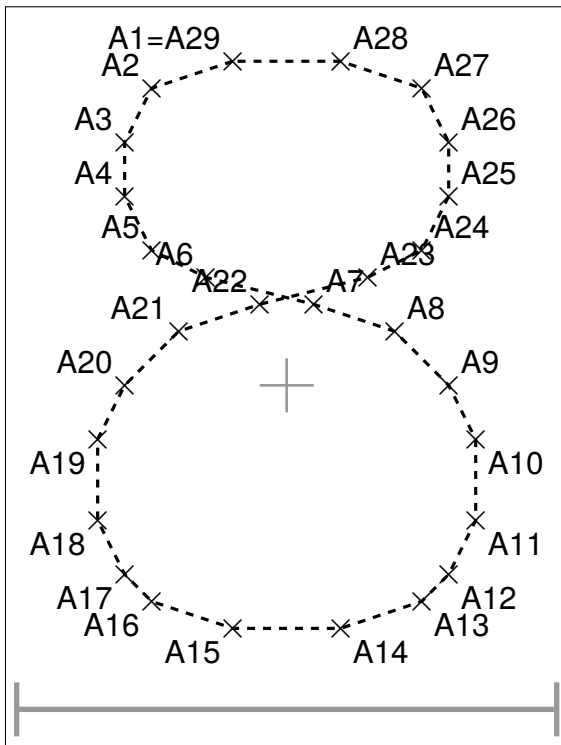
Kuva 34. Numero 5.



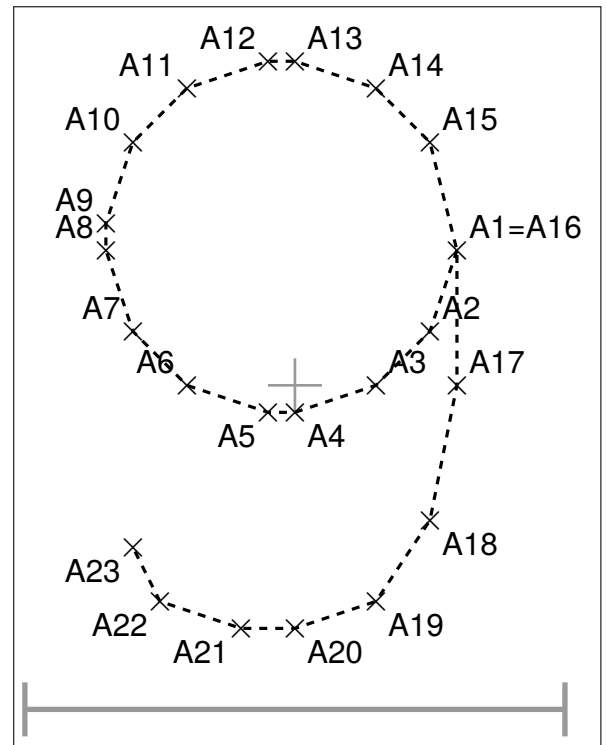
Kuva 35. Numero 6.



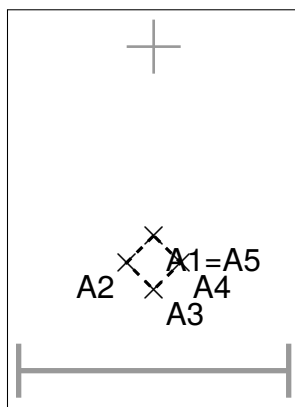
Kuva 36. Numero 7.



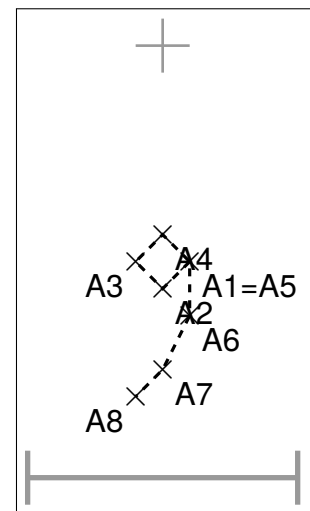
Kuva 37. Numero 8.



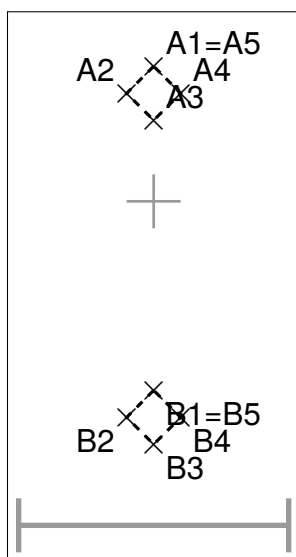
Kuva 38. Numero 9.



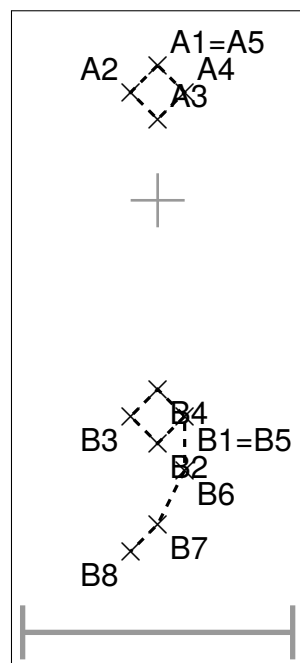
Kuva 39. Piste.



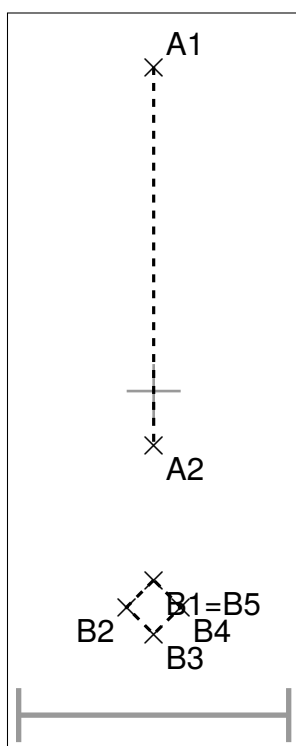
Kuva 40. Pilkku.



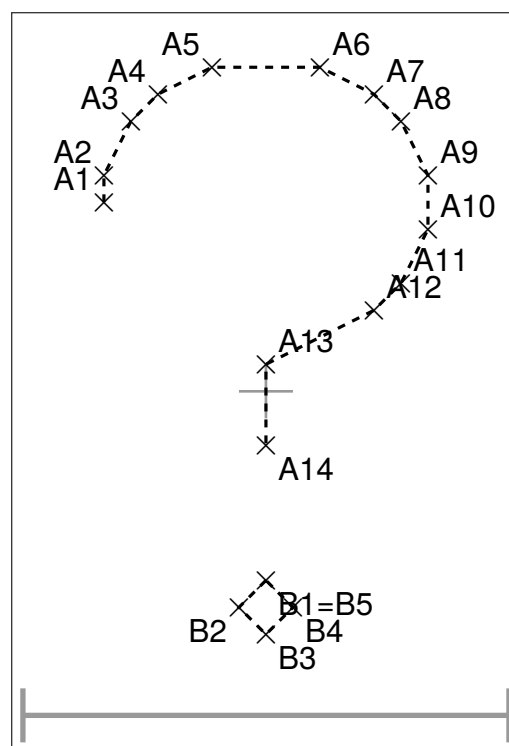
Kuva 41. Kaksoispiste.



Kuva 42. Puolipiste.



Kuva 43. Huutomerkki.



Kuva 44. Kysymysmerkki.

Kirjainluettelot

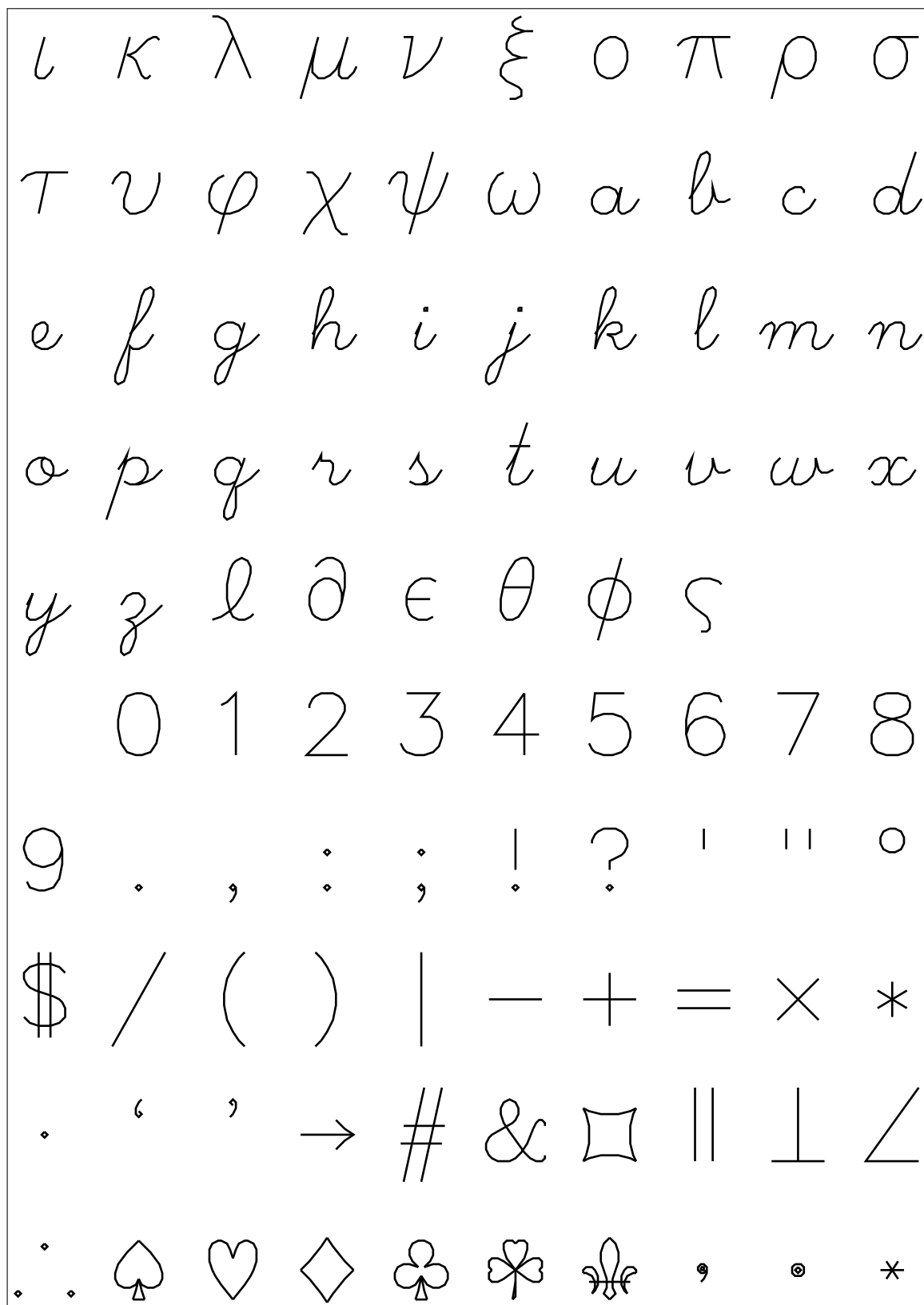
Seuraavassa esitämme luettelon tiedostolistauksessa mukana olevista kirjaimista (kuvat 45–68).

A	B	C	D	E	F	G	H	I	J
K	L	M	N	O	P	Q	R	S	T
U	V	W	X	Y	Z	A	B	Γ	Δ
E	Z	H	Θ	I	K	Λ	M	N	Ξ
Ο	Π	P	Σ	T	Υ	Φ	X	Ψ	Ω
			0	1	2	3	4	5	6
7	8	9	.	,	:	;	!	?	'
"	°	\$	/	()		—	+	=
×	*	.	‘	’	→	#	&	✧	A
B	C	D	E	F	G	H	I	J	K

Kuva 45. Merkit 1–100.



Kuva 46. Merkit 101–200.



Kuva 47. Merkit 201–300.



Kuva 48. Merkit 301–400.

A	B	C	D	E	F	G	H	I	J
K	L	M	N	O	P	Q	R	S	T
U	V	W	X	Y	Z	Α	Β	Γ	Δ
Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ
Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>J</i>
<i>K</i>	<i>L</i>	<i>M</i>	<i>N</i>	<i>O</i>	<i>P</i>	<i>Q</i>	<i>R</i>	<i>S</i>	<i>T</i>
<i>U</i>	<i>V</i>	<i>W</i>	<i>X</i>	<i>Y</i>	<i>Z</i>	a	b	c	d
e	f	g	h	i	j	k	l	m	n
o	p	q	r	s	t	u	v	w	x

Kuva 49. Merkit 401–500.

y	z	α	β	γ	δ	ε	ζ	η	ϑ
ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ
τ	υ	φ	χ	ψ	ω	α	β	γ	δ
ε	φ	g	h	i	j	k	l	m	n
o	p	q	r	s	t	u	v	w	x
y	z	ff	fi	fl	ffi	ffl	l	ε	θ
φ	ς	ff	fi	fl	ffi	ffl	ι		
	0	1	2	3	4	5	6	7	8
9	.	,	:	;	!	?	'	"	°
*	/	()	[]	{	}	<	>

Kuva 50. Merkit 501–600.

		—	+	±	∓	×	·	÷	=
≠	≡	<	>	≤	≥	∞	~	^	′
˘	˘	,	‘	’	ˆ	⊂	⊃	⊄	⊅
∈	→	↑	←	↓	∂	▽	√	∫	ℳ
∞	%	&	@	\$	#	§	†	‡	∃
⊙	♀	♀	⊕	♂	℥	℥	♂	♀	ℙ
☾	♂	*	Ω	♂	Π	Σ	()	[
]	{	}	§	§	√	∫	A	B	C
D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W

Kuva 51. Merkit 601–700.



Kuva 52. Merkit 701–800.



Kuva 53. Merkit 801–900.



Kuva 54. Merkit 901–1000.



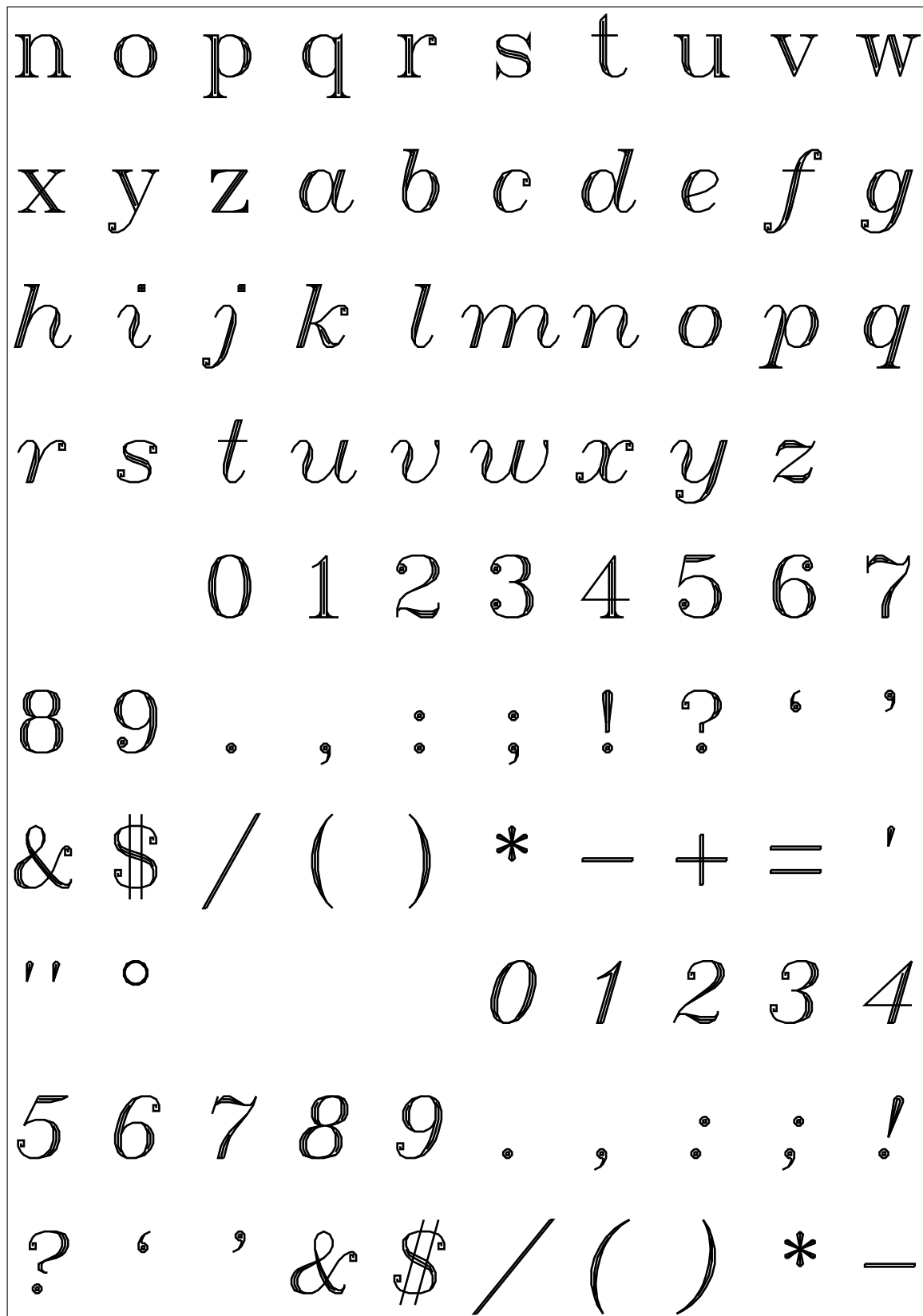
Kuva 55. Merkit 1001–1100.



Kuva 56. Merkit 1101–1200.



Kuva 57. Merkit 1201–1300.



Kuva 58. Merkit 1301–1400.



Kuva 59. Merkit 1401–1500.



Kuva 60. Merkit 1501–1597.

一 万 三 下 五 天 不 可 平 正
百 兩 再 亞 函 画 昼 歪 夏 惡
中 内 甲 本 出 向 曲 果 表 永
氷 半 為 單 九 及 千 午 少 弗
未 年 系 束 卯 来 垂 東 乘 重
島 乙 七 事 二 元 六 市 主 交
京 夜 变 率 商 人 化 今 他 仕
代 合 全 位 伸 体 低 作 余 何
価 例 供 係 値 側 偏 傾 像 億
先 入 八 分 公 共 典 並 其 前

Kuva 61. Oriental-kokoelman merkit 1–100.

円 同 周 次 冷 刀 切 割 力 加
動 北 十 古 直 南 真 幹 準 上
点 反 圧 原 又 双 口 右 号 吸
味 品 哲 嗅 器 四 因 回 凶 固
国 圈 土 去 在 地 均 型 基 塔
場 塩 境 増 士 冬 処 各 夕 外
多 名 大 太 女 如 始 子 存 学
宇 字 宙 定 実 室 家 密 寒 導
小 光 当 常 尺 尾 屈 居 層 山
炭 嵐 川 工 左 項 卷 布 帆 幅

Kuva 62. Oriental-kokoelman merkit 101–200.

干 幾 底 度 庭 座 式 弓 引
弧 弦 強 彈 歸 形 役 徑 彼 後
從 徒 術 御 微 衝 衡 心 性 恒
惑 想 感 態 慣 戈 成 或 戶 扇
手 折 押 持 指 挺 振 捩 接 摘
支 故 教 散 數 文 對 斜 新 方
於 放 日 早 易 昔 明 昨 星 春
時 晶 暑 量 晴 最 暗 曇 題 月
木 析 林 枝 松 柳 柱 相 核 桜
根 械 森 植 極 楫 樂 構 標 橫

Kuva 63. Oriental-kokoelman merkit 201–300.

橋 機 欠 止 此 雌 整 列 死 母
每 比 毛 氏 氤 水 沢 汽 決 波
河 油 法 海 涌 酒 流 液 渦 濕
測 溫 減 淹 源 溶 潮 火 灯 炉
然 燒 無 熱 燐 燃 愛 父 狀 片
牙 牛 物 特 犬 王 玉 珪 球 理
現 瓜 瓦 甘 生 用 田 男 界 思
異 病 癸 白 的 皮 皿 目 具 矛
矢 知 短 石 研 硫 硝 礪 磁 示
利 私 和 秒 科 秋 秤 称 程 稻

Kuva 64. Oriental-kokoelman merkit 301–400.

種 積 穴 究 空 窞 立 竹 第 等
筆 算 管 籠 米 料 粒 粘 糸 級
純 紙 素 組 終 細 經 繪 絕 結
絹 統 綵 緯 線 締 縱 置 羊 美
差 着 羽 翠 翼 老 考 者 耳 取
書 肉 有 胞 期 朝 臣 自 至 舌
乱 辞 舟 般 船 良 色 花 若 草
茶 荷 菊 葉 蒸 藥 虚 虫 血 行
衣 装 複 西 要 見 角 解 言 計
記 証 話 語 讀 誰 論 谷 豆 象

Kuva 65. Oriental-kokoelman merkit 401–500.

貝負質赤走超足距路身
射車軌轉軸輕較輪輻辛
込込近逆速造連通進達
遲過道運遠還酸醜积番
里金鉉鉄銅銀鋼長門問
閉問開関聞限除降陸陰
陽雄難離雨雪雲零雷電
震靈青静非翡面革音類
風飛食首香馬駮骨高鬼
魚鳥鹿麦麻黄黒墨鼓鼻

Kuva 66. Oriental-kokoelman merkit 501–600.

齒 竜 亀 あ い う え お か き
く け こ さ し す せ そ た ち
つ て と な に ぬ ね の は ひ
ふ へ ほ ま み む め も や い
ゆ え よ ら り る れ ろ わ ゐ
う ゑ を ん が ぎ ぐ げ ご ざ
じ ず ぜ ぞ だ ち づ で ど ば
び ぶ べ ぼ ぱ ぴ ぷ ぺ ぽ ア
イ ウ エ オ カ キ ク ケ コ サ
シ ス セ ソ タ チ ツ テ ト ナ

Kuva 67. Oriental-kokoelman merkit 601–700.

ニヌ ネ ノ ハ ヒ フ ヘ ホ マ
ミ ム メ モ ヤ イ ユ エ ヨ ラ
リ ル レ ロ ワ ㇿ ウ エ ヲ ン
ガ ギ グ ゲ ゴ ザ ジ ズ ゼ ゴ
ダ チ ツ テ ト バ ビ ブ ベ ボ
パ ピ プ ペ ポ . , ー

Kuva 68. Oriental-kokoelman merkit 701–758.