

Esercizio

Un formato per il salvataggio di documenti in grafica vettoriale ha una struttura costituita di elementi memorizzati in formato testo nel seguente modo:

- 1) I valori sono memorizzati come `<identificatore><ws><stringa><ws>`, dove `<ws>` è uno o più tra spazio, tab, CR o LF, `<identificatore>` è una sequenza di caratteri non `<ws>`, `<stringa>` è una sequenza di caratteri qualsiasi tra doppi apici. Se in una stringa si trovano due doppi apici di seguito senza nulla in mezzo, vanno sostituiti con un solo doppio apice.

Esempi validi di valori sono:

```
width      "21.0cm"
height "29.7cm"
title      "Titolo della ""mia!"" pagina"
```

In questo esempio il valore di `title` è Titolo della "mia!" pagina.

- 2) Gli oggetti sono memorizzati come `<identificatore><ws>obj<ws>`, seguito da un numero arbitrario di valori o oggetti, seguiti da `<identificatore><ws>end<ws>`. Notare che dentro ogni oggetto possono esserci altri oggetti.

Un esempio valido di oggetto è:

```
ellipse    obj
  cx      "150mm"
  cy      "100mm"
  rx      "200mm"
  ry      "100mm"
  fill    "#FFD700"
ellipse    end
```

Un altro esempio valido è

```
rectangle  obj
  x        "10mm"
  y        "30mm"
  width    "50mm"
  height   "70mm"
  hidden   "true"
  stroke-width "0.1mm"

  transform obj
    translate obj
      x      "-25mm"
      y      "-35mm"
    translate end
  transform end
rectangle  end
```

In questo formato è previsto che un oggetto sia nascosto se contiene il valore `hidden` con valore `true`.

Le immagini possono essere inserite in questo formato in forma testuale non compressa come oggetti `image` che al loro interno contengono un oggetto `data`. L'oggetto `data` deve contenere almeno i tre campi seguenti `width`, il numero di pixel in orizzontale, `height`, il numero di pixel in verticale e `pixel`, le terne di byte RGB codificate con l'encoding Base64 (descritto nel file "From RFC 1521.pdf"). Ad esempio una immagine 3x2 con la prima riga nera e la seconda riga bianca verrebbe codificata come:

```
image obj
  data  obj
    width      "3"
    height     "2"
    pixel      "AAAAAAAAAAAA////////////////"
  data  end
image end
```

Si realizzi un programma che accetti la seguente sintassi:

```
image_extract <input file> <output prefix>
```

Il programma accetta in input un file nel formato indicato, ne estrae la struttura e per ogni immagine non nascosta e non contenuta in oggetti nascosti, la salva in formato PPM. I nomi delle immagini di output saranno dati dal prefisso fornito a linea di comando a cui aggiungere 0001, 0002, ..., 0009, 0010, 0011, 0012, ... e poi l'estensione “.ppm”.