```c
#include <stdio.h>
#include <math.h>
#include <pango/pangocairo.h>
#include <cairo/cairo-ps.h>

// dimensions of the A4 page
//
#define PAGE_WIDTH  595
#define PAGE_HEIGHT 842

#define LINE_SIZE 10
#define TEXT_FONT "Monospace 6"
#define OCRB_FONT "OCRB 6"

void rendertext(cairo_t *cr, char *text);
void renderocrb(cairo_t *cr, char *text);

int main(int argc, char* argv[]) {
        cairo_t *cr;
        cairo_status_t status;
        cairo_surface_t *surface;
        char buf[128];

        sprintf(buf, "<b>TEXT_FONT: [%s] LINE_SIZE: [%d], OCRB_FONT: [%s]</b>",
                TEXT_FONT, LINE_SIZE, OCRB_FONT);

        surface = cairo_ps_surface_create("my1st.ps", PAGE_WIDTH, PAGE_HEIGHT);
        status = cairo_surface_status (surface);
        if (status != CAIRO_STATUS_SUCCESS) {
                // if 'status' was not set to indicate a successful operation, error
                printf("cairo_ps_surface_create() does not give CAIRO_STATUS_SUCCESS\n");
                return 1;
        }
        cr = cairo_create(surface);
        cairo_set_source_rgb(cr, 1.0, 1.0, 1.0);
        cairo_paint(cr);

        cairo_ps_surface_dsc_begin_page_setup (surface);
        cairo_ps_surface_dsc_comment (surface, "%%PageOrientation: Portrait");

        cairo_ps_surface_dsc_comment (surface, "%%Title: My 1st Cairo-PS document");
        cairo_ps_surface_dsc_comment (surface, "%%Copyright: Copyright (C) 2011 guru@unixarea.de");

        cairo_ps_surface_dsc_begin_setup (surface);
        cairo_ps_surface_dsc_comment (surface, "%%IncludeFeature: *MediaColor White");

        cairo_ps_surface_dsc_begin_page_setup (surface);
        cairo_ps_surface_dsc_comment (surface, "%%IncludeFeature: *PageSize A4");
        cairo_ps_surface_dsc_comment (surface, "%%IncludeFeature: *InputSlot LargeCapacity");
        cairo_ps_surface_dsc_comment (surface, "%%IncludeFeature: *MediaType Glossy");
        cairo_ps_surface_dsc_comment (surface, "%%IncludeFeature: *MediaColor Blue");

        rendertext(cr, buf);
        rendertext(cr, "");
        rendertext(cr, "<i>Arabic</i> Ø§Ù□Ø³Ù□Ø§Ù□ Ø¹Ù□Ù□Ù□Ù□ ");
        rendertext(cr, "<b>Bengali</b> (à¦¬à¦¾à§□à¦²à¦¾)     à¦·à¦¾à§à¦□à¦¤à§□à¦®");
        rendertext(cr, "<i><b>Greek</b></i> (Î□Î»Î»Î·Î½Î¹Îº Î¬)  Î□ÎµÎ¹Î¬ Ï□Î±Ï□");
        rendertext(cr, "<u>Hebrew</u> ×©×□Ö¸×□×□Ö¹×□");
        rendertext(cr, "<b>Japanese  (æ□¥æ□¬èª□) ã□□ã□□ã□«ã□¡ã□¯, ï½ºï¾□ï¾□ï¾□ï¾□ </b>");
        rendertext(cr, "Chinese  (ä¸-æ□□,æ□®é□□è¯□,æ±□è¯-) ä½ å¥½");
        rendertext(cr, "Vietnamese      (Tiáº¿ng Viá»□t)        Xin ChÃ o");

        rendertext(cr, "<span font=\"" OCRB_FONT "\">ABC 0123456789</span>");

        rendertext(cr, "EspaÃ±ol Â¡Que vivan los sueÃ±os!");
        rendertext(cr, "normal and OCR on same line: <span font=\"" OCRB_FONT "\">ABC 0123456789</span>");

        cairo_show_page(cr);

        cairo_ps_surface_dsc_comment (surface, "%%IncludeFeature: *PageSize A4");

        cairo_destroy(cr);
        cairo_surface_destroy(surface);

        return 0;
}

static float line = 2*LINE_SIZE;

void rendertext(cairo_t *cr, char *text) {
        PangoLayout *layout;
        PangoFontDescription *desc;

        line = line + LINE_SIZE;
```

```
        layout = pango_cairo_create_layout(cr);

        pango_layout_set_markup(layout, text, -1);

        desc = pango_font_description_from_string( TEXT_FONT );
        pango_layout_set_font_description(layout, desc);
        pango_font_description_free(desc);

        cairo_set_source_rgb(cr, 0.0, 0.0, 1.0);
        pango_cairo_update_layout(cr, layout);

        cairo_move_to(cr, 20, line);

        pango_cairo_show_layout(cr, layout);

        g_object_unref(layout);
}
```