Source code of "dump_analysis.cpp"

```
/*****************************************************************************
Program name:   dump_analysis
Module name:    dump_analysis.cpp
Description:    The program converts "SST 27SF512-5.bin" and "SST 27SF512-4.bin"
                to "program_ROM.txt" and "pattern_ROM.txt".
                Input files :
                 "SST 27SF512-5.bin" ROM dump binary file (high byte)
                 "SST 27SF512-4.bin" ROM dump binary file (low byte)
                Output files :
                 "program_ROM.csv" program ROM code tab-delimited text file
                 "pattern_ROM.csv" pattern ROM code tab-delimited text file
Usage:          dump_analysis <enter>
Version:        1.0
Date:           July, 10, 2021
Programmer:     Tetsuji Oguchi
(C) Oguchi R&D  2021
*****************************************************************************/

#include   <stdio.h>
#include   <string.h>
#include   <process.h>
#include   <clocale>

static char   inhbfile[] = "SST 27SF512-5.bin", inlbfile[] = "SST 27SF512-4.bin";
static char   out1file[] = "program_ROM.csv", out2file[] = "pattern_ROM.csv";
static int    hb, lb;
static int    i, j, r;
static int    pcode, patcode;

FILE* inhbfp, * inlbfp, * txt1ofp, * txt2ofp;

void pat_out()
{
    setlocale(LC_ALL, "en_US.UTF-8");
    wchar_t blank[] = L"\u2000\u2000";
    wchar_t ones[] = L"\u2588\u2588";

    r = fgetc(inhbfp);
    r &= 0x10;
    if (r == 0) fprintf(txt2ofp, "%ls", blank);
    else fprintf(txt2ofp, "%ls", ones);
}

int main()
{
    printf("NEC uPD777 program ROM & pattern ROM dump program 2\n");
    printf("                              (C) Oguchi R&D  2021\n\n");

    // Check & specify file I/O
    if (fopen_s(&inhbfp, "SST 27SF512-5.bin", "rb"))
    {
        printf("Input file SST 27SF512-5.bin doesn't exist\n");
        exit(0);
    }
    if (fopen_s(&inlbfp, "SST 27SF512-4.bin", "rb"))
    {
        printf("Input file SST 27SF512-4.bin open error...\n");
        exit(0);
    }
```

```
    if (fopen_s(&txt1ofp, "program_ROM.csv", "w"))
    {
        printf("Output file program_ROM.csv open error...\n");
        exit(0);
    }
    if (fopen_s(&txt2ofp, "pattern_ROM.csv", "w"))
    {
        printf("Output file pattern_ROM.csv open error...\n");
        exit(0);
    }

    // Start dumped program ROM code conversion
    for (j = 0; j < 16; j++)
    {
        for (i = 0; i < 127; i++)
        {
            hb = fgetc(inhbfp);
            lb = fgetc(inlbfp);
            hb &= 0xf;
            hb <<= 8;
            pcode = hb + lb;
            fprintf(txt1ofp, "%03X\t\n", pcode);
        }
        fprintf(txt1ofp, "\t\n");
    }

    // Start dumped pattern ROM code conversion
    for (j = 0; j < 128; j++)
    {
        for (i = 0; i < 7; i++)
        {
            r = fgetc(inhbfp);
            r = fgetc(inhbfp);
            r = fgetc(inhbfp);
            pat_out();
            pat_out();
            pat_out();
            pat_out();
            pat_out();
            pat_out();
            pat_out();
            pat_out();
            fprintf(txt2ofp, "\t\n");
        }
        fprintf(txt2ofp, "--\t\n");
    }

    // End dump
    printf("NEC uPD777 program ROM & pattern ROM dump completed\n");

    fclose(inhbfp);
    fclose(inlbfp);
    fclose(txt1ofp);
    fclose(txt2ofp);
}
```