```
 Name        : test_IP.c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "800x600.h"
#include "imse_allocation.h"

int main(void)
{
    const int INTER_RESIZE_COEF_BITS = 11;
    const int IMSE_EXTRA_PRECISION_BITS = 4;
    const int IMSE_RESIZE_COEF_SCALE = 1 << (INTER_RESIZE_COEF_BITS +
IMSE_EXTRA_PRECISION_BITS);
    int *TFG_BASE_ADDR = (int *) 0x80000A00;

    /*clock_t t_ini, t_fin;
    double secs;*/

    int src_width = 800;
    int src_height = 600;
    int dst_width = 640;
    int dst_height = 480;
    int shifted_scale_alpha = (IMSE_RESIZE_COEF_SCALE * src_width) / dst_width;
    int shifted_scale_beta = (IMSE_RESIZE_COEF_SCALE * src_height) / dst_height;

    while (TFG_BASE_ADDR [0] == 1)
    {}


    char* data_out = NULL;
    data_out = (char*) imse_Malloc(1024*1024*sizeof(char));
    TFG_BASE_ADDR [1] = 0;                      // reset for new transfer
    TFG_BASE_ADDR [2] = &src_data[0];           // read address
    TFG_BASE_ADDR [3] = &data_out[0];               // write address
    TFG_BASE_ADDR [4] = src_width;              // source row size
    TFG_BASE_ADDR [5] = src_height;             // source col size
    TFG_BASE_ADDR [6] = dst_width;              // destination row size
    TFG_BASE_ADDR [7] = dst_height;             // destination col size
    TFG_BASE_ADDR [8] = shifted_scale_alpha;    // row scale coefficient
    TFG_BASE_ADDR [9] = shifted_scale_beta;     // col scale coefficient


    //t_ini = clock ();
    TFG_BASE_ADDR [1] = 0x1;                        // start and read

    /*for (i = 0; i < 10000000; i++)
        asm ("nop");*/

    while (TFG_BASE_ADDR [0] == 1)
    {}
    /*t_fin = clock ();

    secs = (double) (t_fin - t_ini) / CLOCKS_PER_SEC;

    printf ("%.16g miliseconds.\n", secs * 1000.0);*/

    imse_Free(&data_out);
    return EXIT_SUCCESS;
}
```