Lab 3 - Task 8

To help you along with Lab 3, we are offering you the solution to Task 8.
You can use it as you wish:
- To verify your understanding of Task 8 and of image manipulation
- To verify your own solution to Task 8
- To submit as your solution to Task 8

Make sure you understand what the code does as it may be part of our Lab Exams.

Enjoy!

```c
/* TASK 8 */

// Return a new image of size rows/2 by cols/2 If the original image
// has an odd number of columns, ignore its rightmost column. If the
// original image has an odd number of rows, ignore its bottom row.
// The value of a pixel at (p,q) in the new image is the average of
// the four pixels at (2p,2q), (2p+1,2q), (2p+1,2q+1), (2p,2q+1) in
// the original image.
uint8_t* half( const uint8_t array[],
              unsigned int cols,
              unsigned int rows )
{
    // return array;

    // allocate an image half the original size.
    // note that integer division rounds by truncation towards zero,
    // e.g. 7/2 = 3
    uint8_t *ret = malloc((rows/2)*(cols/2)*sizeof(uint8_t));
    if (ret != NULL)
    {
        // for all pixels in the new, smaller image
        for (unsigned int y = 0; y < rows/2; y++)
            for (unsigned int x = 0; x < cols/2; x++)
            {
                // sum the values of the four pixels in the original image
                // that correspond to this pixel in the new image
                unsigned int total = 0;
                for (unsigned int i = 2*y; i < 2*y+2; i++)
                    for (unsigned int j = 2*x; j < 2*x+2; j++)
                        total += array[i*cols + j];

                // set the new image pixel to the average color
                // ret[y*(cols/2) + x] = (total + 2)/4;
                ret[y*(cols/2) + x] = round(total/4.0);
            }
    }

    return ret;
}```