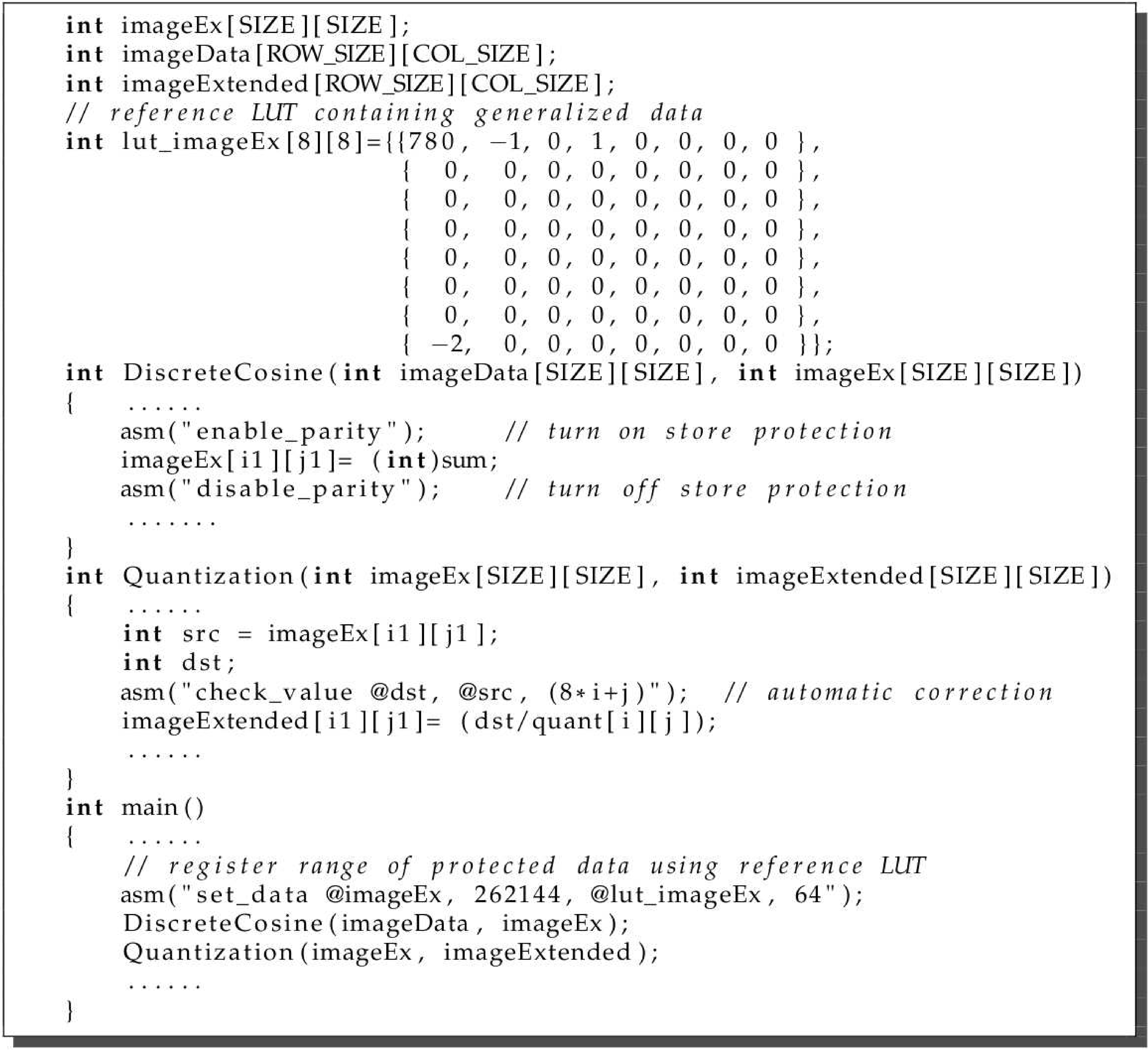

```
int imageEx[SIZE][SIZE];
int imageData[ROW_SIZE][COL_SIZE];
int imageExtended[ROW_SIZE][COL_SIZE];
// reference LUT containing generalized data
int lut_imageEx[8][8]={{780, -1, 0, 1, 0, 0, 0, 0 },
                       {  0,  0, 0, 0, 0, 0, 0, 0 },
                       {  0,  0, 0, 0, 0, 0, 0, 0 },
                       {  0,  0, 0, 0, 0, 0, 0, 0 },
                       {  0,  0, 0, 0, 0, 0, 0, 0 },
                       {  0,  0, 0, 0, 0, 0, 0, 0 },
                       {  0,  0, 0, 0, 0, 0, 0, 0 },
                       { -2,  0, 0, 0, 0, 0, 0, 0 }};
int DiscreteCosine(int imageData[SIZE][SIZE], int imageEx[SIZE][SIZE])
{    ......
    asm("enable_parity");      // turn on store protection
    imageEx[i1][j1]= (int)sum;
    asm("disable_parity");     // turn off store protection
     .......
}
int Quantization(int imageEx[SIZE][SIZE], int imageExtended[SIZE][SIZE])
{    ......
    int src = imageEx[i1][j1];
    int dst;
    asm("check_value @dst, @src, (8*i+j)");  // automatic correction
    imageExtended[i1][j1]= (dst/quant[i][j]);
     ......
}
int main()
{    ......
    // register range of protected data using reference LUT
    asm("set_data @imageEx, 262144, @lut_imageEx, 64");
    DiscreteCosine(imageData, imageEx);
    Quantization(imageEx, imageExtended);
     ......
}
```