The purpose of this project is to investigate properties and applications of Haar wavelets. In particular, methods for compressing audio signals and digital images are investigated.

(1) In the notes, I explain how a vector vector \( u = (u_1, \ldots, u_m) \) corresponds to a piecewise linear function over the interval \([0, 1)\). The function \( \text{plf}(u) \) is defined such that

\[
\text{plf}(u)(x) = u_i, \quad \frac{i-1}{m} \leq x < \frac{i}{m}, \quad 1 \leq i \leq m.
\]

In words, the function \( \text{plf}(u) \) has the value \( u_1 \) on the interval \([0, 1/m)\), the value \( u_2 \) on \([1/m, 2/m)\), etc., and the value \( u_m \) on the interval \([(m-1)/m, 1)\).

Write a \texttt{Matlab} program that takes as input a vector \( u \) and plots the corresponding function \( \text{plf}(u) \). Test your program on several inputs including

\[ u = [0 2 4 6 4 2 1 -1 -2 -4 -6 -6 -4 -2 0], \]

and the vectors \( w \) obtained by concatenating \( u \) with itself 8 and 9 times. Note that \( u \) contains 16 integers, and the last one is 0, not \(-20\) (spaces are important).

(2) Write two \texttt{Matlab} functions \texttt{haar} and \texttt{haar_inv} implementing the method for computing the Haar transform of a vector and the reconstruction of a vector from its Haar coefficients, as described in the notes.

Test your programs on many input, including

\[ u = [0 2 4 6 4 2 1 -1 -2 -4 -6 -6 -4 -2 0] \]

and the string \( w \) from (1). What do you observe?

(3) Write a \texttt{Matlab} function \texttt{haar_step} that performs only \( k \) rounds of averaging and differencing on some input vector \( u \); the function \texttt{haar_step} takes as input \( u \) and \( k \).

Test your program on the vector \( w \) obtained by concatenating \( u \) with itself 8 eight times. Something remarkable happens starting with \( k = 4 \). Explain the behavior that you observe for \( k = 4, 5, 6, 7 \).
Laod the audio file handel using load handel. This file is saved in the variable y. Keep the first 65536 elements of this vector by doing handel = y(1:65536);. To play and hear the music, do sound(handel). Run haar_step on the vector handel for k = 1. Then play the result. What happens. Can you explain it? Do this again for k = 2, 3. What do you observe?

Write a function haarr_inv_step inverting the function haarr_step. The function haarr_inv_step takes as input a vector v and the number of rounds k.

To check that this function is correct, first apply haarr_step and then haarr_inv_step for the same number of steps k. You should get back the orginal vector.

Run haar on the vector handel to get the Haar transform c. Set the detail coefficients to zero by doing c1 = c; c1(32768:end) = 0;. Then apply haarr_inv to c1 to get handel1. Play handel1. What difference to you observe compared to playing handel? Experiment with other compressions of c.

(4) Write two Matlab functions haarr2D and haarr_inv2D implementing the method for computing the Haar transform of a matrix and the reconstruction of an image from its matrix of Haar coefficients, as described in the notes.

Apply the function haarr_inv2D to the matrix

\[
T = \begin{pmatrix}
1212 & -306 & -146 & -54 & -24 & -68 & -40 & 4 \\
30 & 36 & -90 & -2 & 8 & -20 & 8 & -4 \\
-50 & -10 & -20 & -24 & 0 & 72 & -16 & -16 \\
82 & 38 & -24 & 68 & 48 & -64 & 32 & 8 \\
8 & 8 & -32 & 16 & -48 & -48 & -16 & 16 \\
20 & 20 & -56 & -16 & -16 & 32 & -16 & -16 \\
-8 & 8 & -48 & 0 & -16 & -16 & -16 & -16 \\
44 & 36 & 0 & 8 & 80 & -16 & -16 & 0 \\
\end{pmatrix}
\]

Compare your result with the matrix P of Example 4.1 of the paper by Greg Ames (see the web page for CIS515). The matrix in Ames's paper seems to have at typo! What is it?

You can load and display various images in Matlab using the following lines of code:

clear X map
load('durer','X')
Xdurer = X(1:512,:);
Xdurer(:,510:512) = 50;
figure
colormap(gray)
imagesc(Xdurer)

The above loads the file durer. There are a few other images such as detail, flujet, earth, mandrill, spine, and clown. You may have to resize these images to have dimensions that are powers of 2. To display an image, use imagesc.
Convert Xdurer to its Haar transform and decode it. Compare the original and the reconstructed image.

(5) Write two Matlab functions `haar2D_n` and `haar-inv2D_n` implementing the method for computing the normalized Haar transform of a matrix and the reconstruction of an image from its matrix of normalized Haar coefficients, as described in the notes.

Consider the image given by the following matrix:

\[
A = \begin{bmatrix}
100 & 103 & 99 & 97 & 93 & 94 & 78 & 73 \\
102 & 97 & 100 & 111 & 104 & 96 & 82 \\
99 & 109 & 104 & 95 & 93 & 92 & 88 & 76 \\
114 & 104 & 99 & 102 & 93 & 82 & 74 & 74 \\
96 & 91 & 91 & 87 & 79 & 78 & 77 & 76 \\
90 & 88 & 83 & 78 & 77 & 74 & 76 & 76 \\
92 & 81 & 73 & 72 & 69 & 65 & 66 & 62 \\
75 & 70 & 69 & 65 & 60 & 55 & 61 & 65
\end{bmatrix}
\]

Use `haar2D_n` to compute the normalized matrix \( C \) of Haar coefficients of \( A \).

It is claimed in Ames’s paper (Section 7) that the reconstructed matrix

\[
A_2 = \begin{bmatrix}
100 & 100 & 95 & 95 & 92 & 92 & 76 & 76 \\
103 & 103 & 98 & 98 & 106 & 106 & 90 & 90 \\
99 & 109 & 99 & 99 & 96 & 96 & 81 & 81 \\
114 & 104 & 104 & 91 & 91 & 76 & 76 & 76 \\
91 & 91 & 86 & 86 & 76 & 76 & 76 & 76 \\
91 & 91 & 86 & 86 & 76 & 76 & 76 & 76 \\
82 & 82 & 76 & 76 & 66 & 66 & 66 & 66 \\
74 & 74 & 69 & 69 & 58 & 58 & 59 & 59
\end{bmatrix}
\]

is obtained from the normalized matrix

\[
C_1 = \begin{bmatrix}
255 & 52 & 15 & 21 & 0 & 0 & 0 & 0 \\
78 & 0 & 0 & 22 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
38 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 11 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
15 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

but this not quite correct. First, the coefficient 255 should be 682, and other nonzero entries are missing. Find the matrix \( C_2 \), a compressed version of \( C \), that gives back \( A_2 \).

*Hint.* The command `round` is helpful.
(6) **Extra Credit.** Write versions of `haar2D` and `haar_inv2D` that perform only $k$ rounds of averaging and differencing. Test your programs on `Xdurer` (and possibly other images).

**TOTAL:** 300 + 50 points.