EXPERIMENT-2

Theme: Downsample and Upsample

Name: Anupama Kulshreshtha

Roll no.: EE22BTECH11009

Downsample and upsample the signal given below with a factor 2 and 3 in Matlab and C.

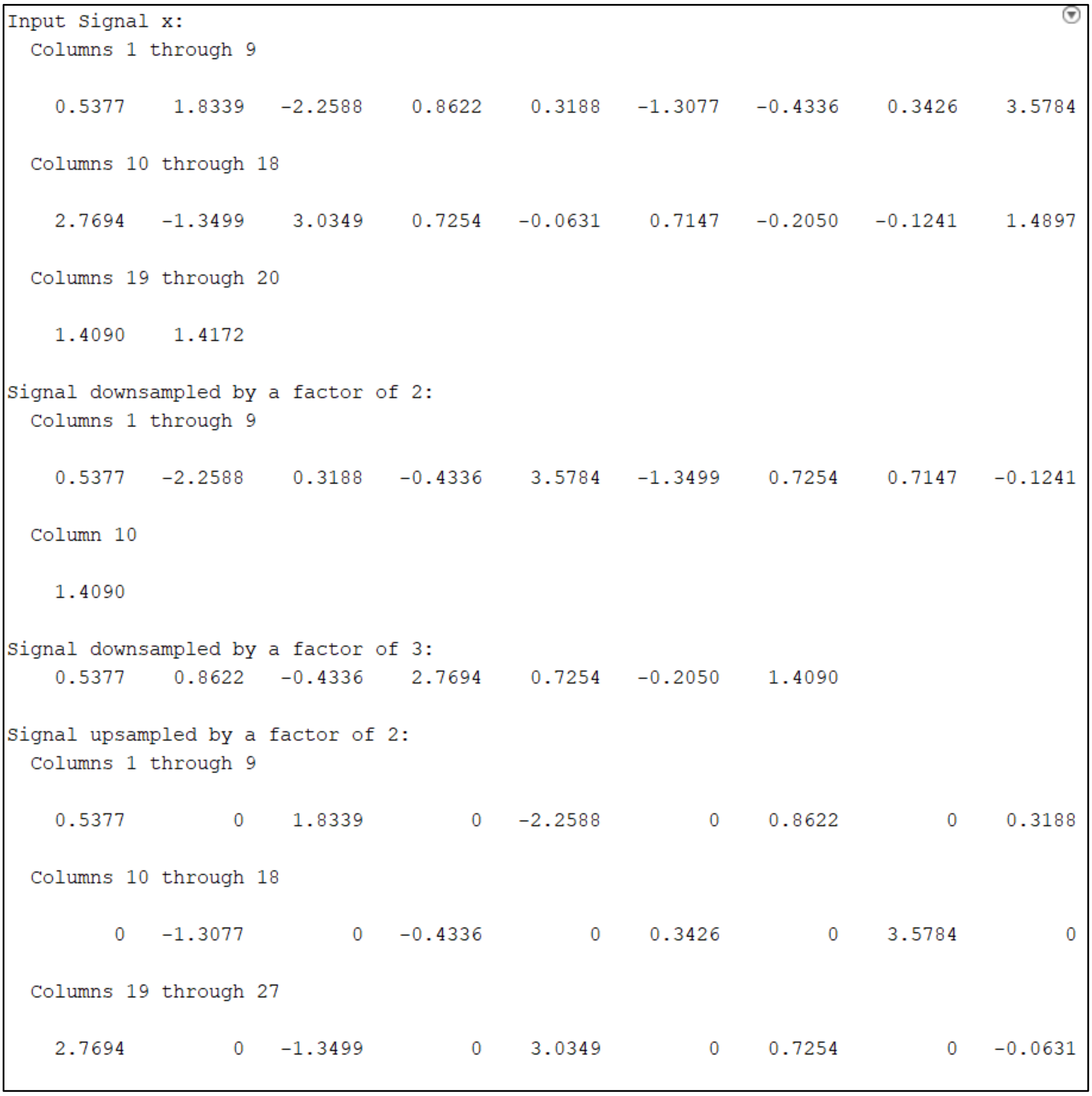
x[n]={0.5377 1.8339 -2.2588 0.8622 0.3188 -1.3077 -0.4336 0.3426 3.5784 2.7694 -1.3499 3.0349 0.7254 -0.0631 0.7147 -0.2050 -0.1241 1.4897 1.4090 1.4172}

(a) MATLAB:

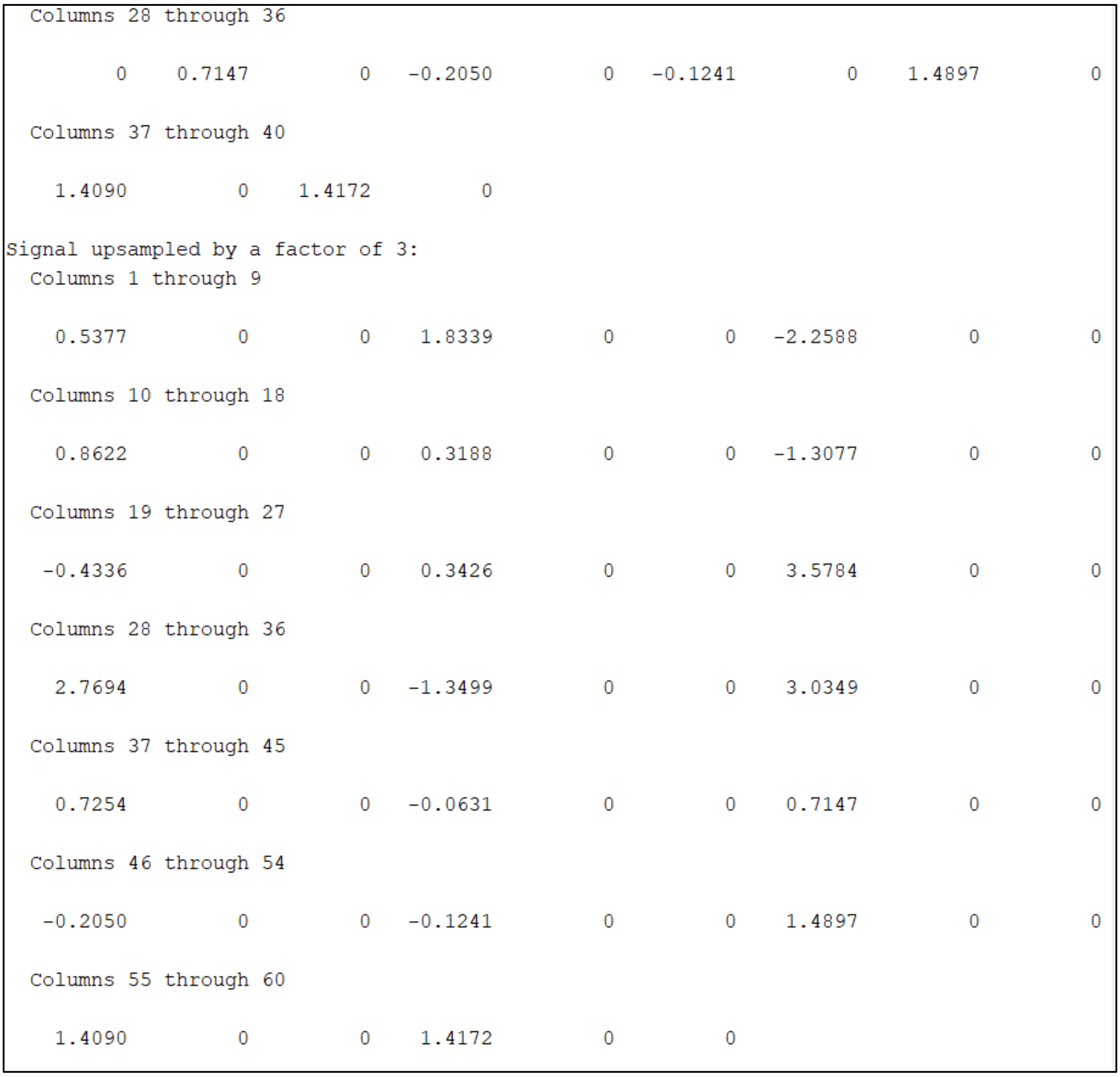
Code:

```
clc
clear
close all

% Define signal
x = [0.5377    1.8339   -2.2588    0.8622    0.3188   -1.3077   -0.4336    0.3426
3.5784    2.7694   -1.3499    3.0349    0.7254   -0.0631    0.7147   -0.2050   -
0.1241    1.4897    1.4090    1.4172];

% Display the input and output signals
disp('Input Signal x:')
disp(x)
disp('Signal downsampled by a factor of 2:')
disp(downsample(x, 2))
disp('Signal downsampled by a factor of 3:')
disp(downsample(x, 3))
disp('Signal upsampled by a factor of 2:')
disp(upsample(x, 2))
disp('Signal upsampled by a factor of 3:')
disp(upsample(x, 3))

function result = downsample(x, m)
    % Lengths of the signal
    L = length(x);
    k = round(L/m);
    % Initialize the result signal
    result = zeros(1, k);
    result(1) = x(1);

    for i = 2:k
        if m*(i-1)+1 <= L
        result(i) = x(m*(i-1)+1);
        end
    end
end

function result = upsample(x,m)
    % Lengths of the signal
    L = length(x);
    k = m*L;
    % Initialize the result signal
    result = zeros(1, k);
    result(1) = x(1);

    for i = 2:k
        if mod(i-1,m) == 0
            result(i) = x(((i-1)/m)+1);
        else
            result(i) = 0;
        end
    end
end
```

Output:



```
Input Signal x:
  Columns 1 through 9

    0.5377    1.8339   -2.2588    0.8622    0.3188   -1.3077   -0.4336    0.3426    3.5784

  Columns 10 through 18

    2.7694   -1.3499    3.0349    0.7254   -0.0631    0.7147   -0.2050   -0.1241    1.4897

  Columns 19 through 20

    1.4090    1.4172

Signal downsampled by a factor of 2:
  Columns 1 through 9

    0.5377   -2.2588    0.3188   -0.4336    3.5784   -1.3499    0.7254    0.7147   -0.1241

  Column 10

    1.4090

Signal downsampled by a factor of 3:
    0.5377    0.8622   -0.4336    2.7694    0.7254   -0.2050    1.4090

Signal upsampled by a factor of 2:
  Columns 1 through 9

    0.5377         0    1.8339         0   -2.2588         0    0.8622         0    0.3188

  Columns 10 through 18

         0   -1.3077         0   -0.4336         0    0.3426         0    3.5784         0

  Columns 19 through 27

    2.7694         0   -1.3499         0    3.0349         0    0.7254         0   -0.0631
```



```
  Columns 28 through 36

         0    0.7147         0   -0.2050         0   -0.1241         0    1.4897         0

  Columns 37 through 40

    1.4090         0    1.4172         0

Signal upsampled by a factor of 3:
  Columns 1 through 9

    0.5377         0         0    1.8339         0         0   -2.2588         0         0

  Columns 10 through 18

    0.8622         0         0    0.3188         0         0   -1.3077         0         0

  Columns 19 through 27

   -0.4336         0         0    0.3426         0         0    3.5784         0         0

  Columns 28 through 36

    2.7694         0         0   -1.3499         0         0    3.0349         0         0

  Columns 37 through 45

    0.7254         0         0   -0.0631         0         0    0.7147         0         0

  Columns 46 through 54

   -0.2050         0         0   -0.1241         0         0    1.4897         0         0

  Columns 55 through 60

    1.4090         0         0    1.4172         0         0
```

(b) C:

Code:

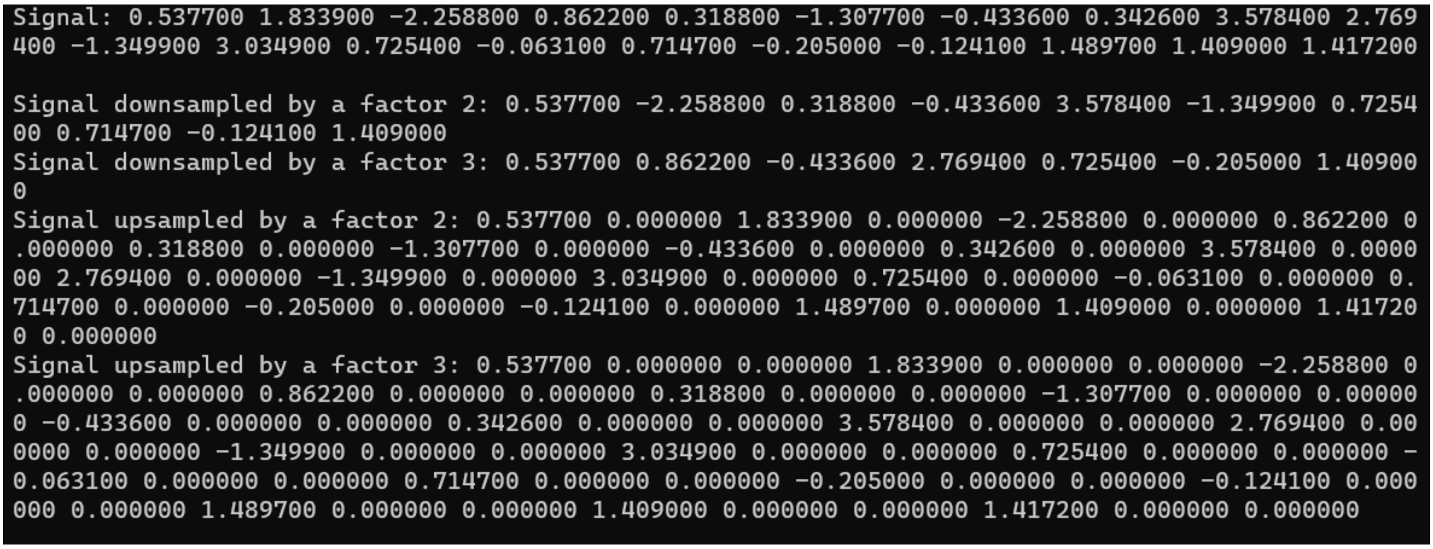
File: main.c

```
#include <stdio.h>

#include <stdlib.h>


int main()

{

    double signal[] = {0.5377,  1.8339,  -2.2588,  0.8622,  0.3188, -1.3077, -0.4336, 0.3426,  3.5784,  2.7694, -1.3499,  3.0349,  0.7254, -0.0631,  0.7147,  -0.2050, -0.1241,  1.4897,  1.4090,  1.4172};

    int L = sizeof(signal)/sizeof(signal[0]);
    double downsample_2[downsample_length (L, 2)];
    double downsample_3[downsample_length (L, 3)];

    double upsample_2[2*L];
    double upsample_3[3*L];

    downsample(signal, L, 2, downsample_2);
    downsample(signal, L, 3, downsample_3);
    upsample(signal, L, 2, upsample_2);
    upsample(signal, L, 3, upsample_3);

    printf("Signal: ");
    printArray(signal, L);

    printf("Signal downsampled by a factor 2: ");
    printArray(downsample_2, downsample_length (L, 2));

    printf("Signal downsampled by a factor 3: ");
    printArray(downsample_3, downsample_length (L, 3));

    printf("Signal upsampled by a factor 2: ");
    printArray(upsample_2, 2*L);

    printf("Signal upsampled by a factor 3: ");
    printArray(upsample_3, 3*L);
    return 0;
}
```

File: functions.c

```
#include <stdio.h>
#include <math.h>

int downsample_length (int L, int m){
    int k = L/m;
    if (L % m == 0){
        k = L/m;
    }
    else{
        k = k+1;
    }
    return k;
}
void downsample(double *x, int L, int m, double *result){
    int k = downsample_length(L, m);
    for(int i=0; i<k; i++){
        result[i] = 0;
    }
    for(int i=0; i<k; i++){
        if (m*i < L){
            result[i] = x[m*i];
        }
    }
}

void upsample(double *x, int L, int m, double *result){
    int k = m*L;
    for(int i=0; i<k; i++){
        result[i] = 0;
    }
    for (int i=0; i<k; i++){
        if(i%m==0){
            result[i] = x[i/m];
        }
        else{
            result[i] = 0;
        }
    }
}

void printArray(double array[], int size){
    for(int i=0; i<size; i++){
        printf("%f ", array[i]);
    }
    printf("\n");
}
```

Output:



```
Signal: 0.537700 1.833900 -2.258800 0.862200 0.318800 -1.307700 -0.433600 0.342600 3.578400 2.769
400 -1.349900 3.034900 0.725400 -0.063100 0.714700 -0.205000 -0.124100 1.489700 1.409000 1.417200

Signal downsampled by a factor 2: 0.537700 -2.258800 0.318800 -0.433600 3.578400 -1.349900 0.7254
00 0.714700 -0.124100 1.409000
Signal downsampled by a factor 3: 0.537700 0.862200 -0.433600 2.769400 0.725400 -0.205000 1.40900
0
Signal upsampled by a factor 2: 0.537700 0.000000 1.833900 0.000000 -2.258800 0.000000 0.862200 0
.000000 0.318800 0.000000 -1.307700 0.000000 -0.433600 0.000000 0.342600 0.000000 3.578400 0.0000
00 2.769400 0.000000 -1.349900 0.000000 3.034900 0.000000 0.725400 0.000000 -0.063100 0.000000 0.
714700 0.000000 -0.205000 0.000000 -0.124100 0.000000 1.489700 0.000000 1.409000 0.000000 1.41720
0 0.000000
Signal upsampled by a factor 3: 0.537700 0.000000 0.000000 1.833900 0.000000 0.000000 -2.258800 0
.000000 0.000000 0.862200 0.000000 0.000000 0.318800 0.000000 0.000000 -1.307700 0.000000 0.00000
0 -0.433600 0.000000 0.000000 0.342600 0.000000 0.000000 3.578400 0.000000 0.000000 2.769400 0.00
0000 0.000000 -1.349900 0.000000 0.000000 3.034900 0.000000 0.000000 0.725400 0.000000 0.000000 -
0.063100 0.000000 0.000000 0.714700 0.000000 0.000000 -0.205000 0.000000 0.000000 -0.124100 0.000
000 0.000000 1.489700 0.000000 0.000000 1.409000 0.000000 0.000000 1.417200 0.000000 0.000000
```

Result:

Therefore, the signal

x[n]={0.5377 1.8339 -2.2588 0.8622 0.3188 -1.3077 -0.4336 0.3426 3.5784 2.7694 -1.3499 3.0349 0.7254 -0.0631 0.7147 -0.2050 -0.1241 1.4897 1.4090 1.4172}

gives the following outputs in MATLAB and C.

(a) **Signal downsampled by a factor 2:** {0.537700, -2.258800, 0.318800, -0.433600, 3.578400, -1.349900, 0.725400, 0.714700, -0.124100, 1.409000}

(b) **Signal downsampled by a factor 3:** {0.537700, 0.862200, -0.433600, 2.769400, 0.725400, -0.205000, 1.409000}

(c) **Signal upsampled by a factor 2:** {0.537700, 0.000000, 1.833900, 0.000000, -2.258800, 0.000000, 0.862200, 0.000000, 0.318800, 0.000000, -1.307700, 0.000000, -0.433600, 0.000000, 0.342600, 0.000000, 3.578400, 0.000000, 2.769400, 0.000000, -1.349900, 0.000000, 3.034900, 0.000000, 0.725400, 0.000000, -0.063100, 0.000000, 0.714700, 0.000000, -0.205000, 0.000000, -0.124100, 0.000000, 1.489700, 0.000000, 1.409000, 0.000000, 1.417200, 0.000000}

(d) **Signal upsampled by a factor 3:** {0.537700, 0.000000, 0.000000, 1.833900, 0.000000, 0.000000, -2.258800, 0.000000, 0.000000, 0.862200, 0.000000, 0.000000, 0.318800, 0.000000, 0.000000, -1.307700, 0.000000, 0.000000, -0.433600, 0.000000, 0.000000,

```
0.342600, 0.000000, 0.000000, 3.578400, 0.000000, 0.000000, 2.769400, 0.000000,
0.000000, -1.349900, 0.000000, 0.000000, 3.034900, 0.000000, 0.000000, 0.725400,
0.000000, 0.000000, -0.063100, 0.000000, 0.000000, 0.714700, 0.000000, 0.000000,
-0.205000, 0.000000, 0.000000, -0.124100, 0.000000, 0.000000, 1.489700, 0.000000,
0.000000, 1.409000, 0.000000, 0.000000, 1.417200, 0.000000, 0.000000}
```