

NAME- AMAN KUMAR
ROLL NO- EE22BTECH11006

EXPERIMENT -1
DSP-LAB

1) CONVOLUTION CODE

MATLAB:

```
1. x = [0.3426, 3.5784, 2.7694, -1.3499, 3.0349, 0.7254, -0.0631];
2. h = [0.7147, -0.2050, -0.1241, 1.4897, 1.4090];
3.
4. ans_conv = conv_calculator(x, h);
5. disp('the output is ');
6. disp(ans_conv);
7.
8. function ans_conv = conv_calculator(x, h)
9.     n = length(x);
10.    m = length(h);
11.    N = [x, zeros(1, m - 1)];
12.    M = [h, zeros(1, n - 1)];
13.    Y = zeros(1, m + n - 1);
14.
15.    for i = 1:m + n - 1
16.        for j = 1:n
17.            if (i - j + 1 > 0 && i - j + 1 <= m)
18.                Y(i) = Y(i) + N(j) * M(i - j + 1);
19.            end
20.        end
21.    end
22.    ans_conv = Y;
23. end
```

OUTPUT:

>> conv_func

the output is

Columns 1 through 10

0.2449 2.4872 1.2032 -1.4662 7.9156 9.2314 1.3207 2.5420 5.3646 0.9281

Column 11

-0.0889

C CODE:

```
1. #include <stdio.h>
2.
3. #define MAX_SIZE 100
4.
```

```

5. void convolution(double sig_1[], int m, double sig_2[], int n, double output[]){
6.     for(int i=0; i<m; i++){
7.         for(int j=0; j<n; j++){
8.             output[i+j] += sig_1[i] * sig_2[j];
9.         }
10.    }
11. }
12. int main() {
13.     int m, n;
14.
15.     double sig_1[] = {0.3426, 3.5784, 2.7694, -1.3499, 3.0349, 0.7254, -0.0631};
16.     double sig_2[] = {0.7147, -0.2050, -0.1241, 1.4897, 1.4090};
17.
18.     m = sizeof(sig_1) / sizeof(sig_1[0]);
19.     n = sizeof(sig_2) / sizeof(sig_2[0]);
20.     double output[MAX_SIZE + MAX_SIZE - 1] = {0};
21.
22.     convolution(sig_1, m, sig_2, n, output);
23.
24.     printf("The final output sequence:\n");
25.     for (int i = 0; i < m + n - 1; ++i) {
26.         printf("%1f ", output[i]);
27.     }
28.
29.     return 0;
30. }

```

OUTPUT:

The final output sequence:

```

0.244856 2.487249 1.203202 -1.466209 7.915556 9.231352 1.320703 2.541995 5.364633
0.928089 -0.088908

```

2) CORRELATION CODE:

MATLAB

```

1. x = [0.3426, 3.5784, 2.7694, -1.3499, 3.0349, 0.7254, -0.0631];
2. h = [0.7147, -0.2050, -0.1241, 1.4897, 1.4090];
3.
4. ans_corre = corre_calculator(x, h);
5. disp('the output is ');
6. disp(ans_corre);
7.
8. function ans_corre = corre_calculator(x, h)
9.     n = length(x);
10.    m = length(h);
11.    N = [x, zeros(1, m - 1)];
12.    M = [fliplr(h), zeros(1, n - 1)];
13.    Y = zeros(1, m + n - 1);
14.
15.    for i = 1:m + n - 1
16.        for j = 1:n
17.            if (i - j + 1 > 0 && i - j + 1 <= m)
18.                Y(i) = Y(i) + N(j) * M(i - j + 1);
19.            end
20.        end
21.    end

```

```
22.     ans_corre = Y;  
23. end
```

OUTPUT:

>> correlation

the output is

Columns 1 through 10

0.4827 5.5523 9.1903 1.7093 1.4328 7.7005 2.8711 -1.7710 2.0282 0.5314

Column 11

-0.0451

C CODE:

```
#include <stdio.h>  
  
#define MAX_SIZE 100  
  
void correlation(double sig_1[], int m, double sig_2[], int n, double output[]) {  
    for (int i = 0; i < m; i++) {  
        for (int j = 0; j < n; j++) {  
            output[i + j] += sig_1[i] * sig_2[n - j - 1];  
        }  
    }  
}  
  
int main() {  
    int m, n;  
    double sig_1[] = {0.3426, 3.5784, 2.7694, -1.3499, 3.0349, 0.7254, -0.0631};  
    double sig_2[] = {0.7147, -0.2050, -0.1241, 1.4897, 1.4090};  
  
    m = sizeof(sig_1) / sizeof(sig_1[0]);  
    n = sizeof(sig_2) / sizeof(sig_2[0]);  
  
    double output[MAX_SIZE + MAX_SIZE - 1] = {0};  
  
    correlation(sig_1, m, sig_2, n, output);  
  
    printf("The final output sequence:\n");  
    for (int i = 0; i < m + n - 1; ++i) {  
        printf("%lf ", output[i]);  
    }  
  
    return 0;  
}
```

OUTPUT:

The final output sequence:

0.482723 5.552337 9.190310 1.709254 1.432830 7.700457 2.871109 -1.770950 2.028167 0.531379

0.045098