Shubham Golwal

2020300015

**Experiment 2**

# Convolution Time Domain Operation

**Aim:** To obtain Convolution, Circular Convolution and Linear Convolution using Circular Convolution of L point sequence x[n] and M point sequence h[n] using C language.

**Objective (A):** To obtain Convolution, of L point sequence x[n] and M point sequence h[n].

**Code:**

import java.util.Scanner;

public class Main { public static void main(String[] args) {

System.out.println("Linear Convolution");

Scanner input = new Scanner(System.in); System.out.print("\nEnter the length of x[n]: "); int x\_length = input.nextInt(); // length of x[n] float[] x = new float[x\_length];

System.out.printf("\nEnter the values of x[%d]: ", x\_length);

for (int i = 0; i < x\_length; i++) { x[i] = input.nextFloat(); // values of x[n]

}

System.out.print("\nEnter the length of h[n]: "); int h\_length = input.nextInt(); // length of h[n] float[] h = new float[h\_length];

System.out.printf("\nEnter the values of h[%d]: ", h\_length);

for (int i = 0; i < h\_length; i++) {

h[i] = input.nextFloat(); // values of h[n]

}

int y\_length = x\_length + h\_length - 1; // length of y[n] = L1+L2-1 float[] y = new float[y\_length]; for (int i = 0; i < y\_length; i++) {

y[i] = 0;

}

int offset = 0; for (int i = 0; i < h\_length; i++) { for (int j = 0; j < x\_length; j++) { // Multiplication, addition and shifting

y[j + offset] += h[i] \* x[j];

}

offset++;

}

System.out.printf("\nValues of z[%d] are: ", y\_length);

for (int i = 0; i < y\_length; i++) { // print y[n]

System.out.printf("%f ", y[i]);

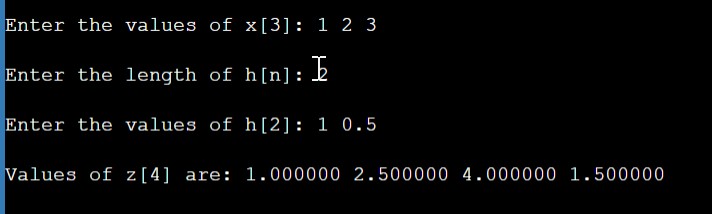
}

System.out.println();

}

}

**Output:**



**Objective (B):** To obtain Circular Convolution of L point sequence x[n] and M point sequence h[n] using.

**Code:**

import java.util.Scanner; public class Main {

public static void main(String[] args) { Scanner input = new Scanner(System.in); int m, n, i, j, k; int[] x, h, y, x2, a;

System.out.print("Enter the length of the first sequence x1[n]: "); m = input.nextInt(); x = new int[m]

System.out.print("Enter the first sequence x1[n]: "); for(i = 0; i < m; i++) { x[i] = input.nextInt();

}

System.out.print("Enter the length of the second sequence x2[n]: "); n = input.nextInt(); h = new int[n];

System.out.print("Enter the second sequence x2[n]: "); for(j = 0; j < n; j++) { h[j] = input.nextInt();

}

if(m != n) { if(m > n) {

h = padWithZero(h, m - n); n = m; } else {

1. = padWithZero(x, n - m); m = n;

}

1. = new int[n]; a = new int[n]; y[0] = 0; a[0] = h[0]; for(i = m; i < n; i++) { x[i] = 0;

}

for(j = 1; j < n; j++) { a[j] = h[n - j];

}

for(i = 0; i < n; i++) { y[0] += x[i] \* a[i];

}

for(k = 1; k < n; k++) { y[k] = 0; x2 = new int[n]; for(j = 1; j < n; j++) { x2[j] = a[j - 1]; } x2[0] = a[n - 1]; for(i = 0; i < n; i++) { a[i] = x2[i]; y[k] += x[i] \* x2[i];

}

} System.out.print("The circular convolution is: "); for(i = 0; i < n; i++) {

System.out.print(y[i] + "\t");

}

}

}

// method to pad an array with zero

private static int[] padWithZero(int[] arr, int padding) { int[] newArr = new int[arr.length + padding]; for(int i = 0; i < arr.length; i++) { newArr[i] = arr[i];

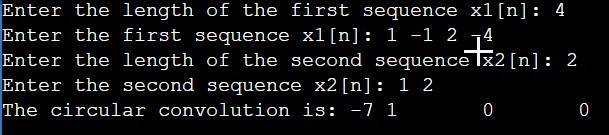
}

return newArr;

}

}

**Output:**



**Objective (C):** To obtain Linear Convolution using Circular Convolution of L point sequencex[n] and M point sequence h[n].

**Code:**

import java.util.Scanner; public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Linear Convolution Using Circular Convolution"); System.out.print("Enter the length of x[n]: "); int m = scanner.nextInt(); // length of x[n] System.out.print("Enter the length of h[n]: "); int n = scanner.nextInt(); // length of h[n] int r = m + n - 1; // length of y[n] = L1+L2-1

System.out.printf("The length of y[n] will be (L+M-1): %d\n", r); int[] x1 = new int[r]; int[] x2 = new int[r]; int[] y = new int[r];

System.out.printf("Enter the values of x1[%d]: ", m); for (int i = 0; i < m; i++) {

x1[i] = scanner.nextInt(); // values of x[n]

}

System.out.printf("Enter the values of x2[%d]: ", n); for (int i = 0; i < n; i++) {

x2[i] = scanner.nextInt(); // values of h[n]

}

// PADDING LOGIC

for (int i = n; i < r; i++) { x2[i] = 0;

}

for (int i = m; i < r; i++) { x1[i] = 0;

}

System.out.print("\nThe input sequence is x[n]: "); for (int i = 0; i < r; i++) {

System.out.print(x1[i] + " ");

}

System.out.println();

System.out.print("\nThe impulse sequence is h[n]: "); for (int i = 0; i < r; i++) {

System.out.print(x2[i] + " ");

}

System.out.println();

// LOGIC OF THE CODE USING ANALYTICAL FORMULA for (int i = 0; i < r; i++) { for (int j = 0; j < r - 1; j++) { if (i - j < 0) {

y[i] += x1[j] \* x2[r + i - j];

} else {

y[i] += x1[j] \* x2[i - j];

}

}

}

// display output sequence

System.out.print("\nThe output sequence y[n]: "); for (int i = 0; i < r; i++) {

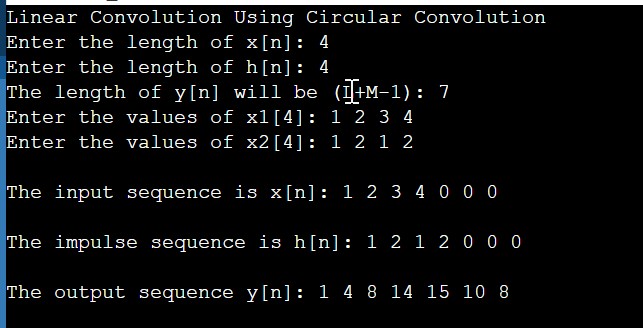
System.out.print(y[i] + " ");

}

System.out.println();

}}

**Output:**



**MATLAB CODE**

%Program to find convulation

clear all; close all; clc;

x=[1,2,3,4]; y=[1,2,1] z=conv(x,y) display(z); z=cconv(x,y,length(x)) display(z);

**Output:**

1. =

* 1. 2 1

1. =

* 1. 4 8 12 11 4

z =

1 4 8 12 11 4

z =

12 8 8 12

z =

12 8 8 12