Name: Shubham Golwal

UID: 2020300015

Batch: TE COMPS-4

EXPERIMENT 6

Fast Fourier Transform

AIM

To obtain Fast Fourier Transform (DFT) of the given L point sequence x[n] using C language and Matlab

CODE

```
#include<stdio.h>
#include<math.h>                 //Include math header file
#define max 4                    //Macro max
void Point4 (int N, float x[4][2], float t[4][2]);   //Function declaration for
4pt DITFFT void Point8 (int N, float x[8][2], float t[8][2]);      //Function
declaration for 8pt DITFFT void  main ()
{


int i, j, k, n, L, N;


float x[8][2], X[8][2], t[8][2]; // declaring arrays to store
//
initialisation
of array   for
(i = 0; i <
max; i++)


   {


X[i][0] = 0;

X[i][1] = 0;
```

```
    x[i][0] = 0;
```

```
x[i][1] = 0;
```

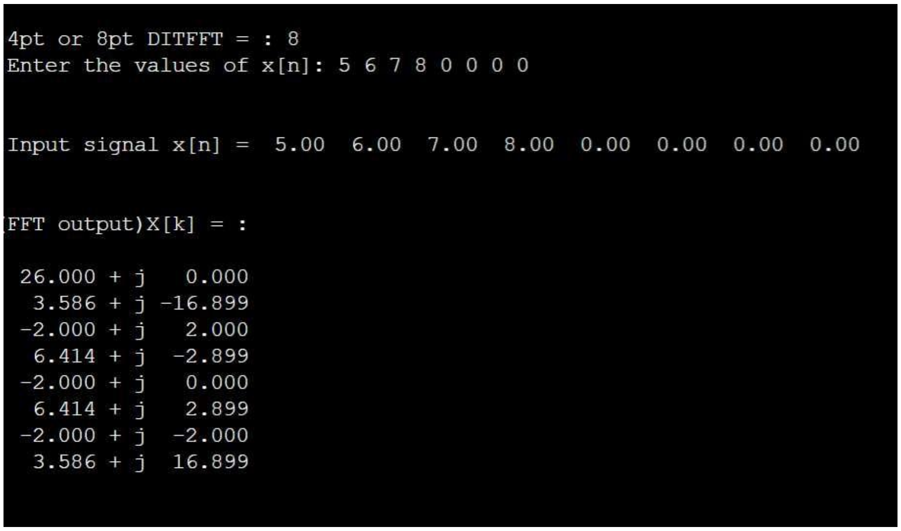
```
}


printf ("\n\n\n 4pt or 8pt DITFFT = : ");


scanf ("%d", &L);


// N must be Radix 2 Number for FFT algorithm
if (L > 4)

N = 8;

  else

N = 4;


printf (" Enter the values of x[n]: ");


for (i = 0; i < L; i++)


    {

scanf ("%f", &x[i][0]);          //assigning input values to the array
    }


printf ("\n\n Input signal x[n] = ");


for (i = 0; i < L; i++)


printf (" %4.2f ", x[i][0]);      //Printing the entered values
// DITFFT operation
if (N == 4)
```

```
Point4 (N, x, X);        // function call for 4pt ditfft

  else if (N == 8)

Point8 (N, x, X);                        // function call for 8pt ditfft
printf ("\n\n\n(FFT output)X[k] = :\n ");

for (k = 0; k < N; k++)

printf ("\n %7.3f + j %7.3f", X[k][0], X[k][1]);

printf ("\n\n");

}



void
Point4 (int N, float x[4][2], float t[4][2])
{
   int a, b, c, d, i, j, k,
n;

float e;

float G[4][2], H[4][2];

for (n = 0; n < N; n++)           //initialisation
    {
t[n][0] = 0;
t[n][1] = 0;

G[n][0] = 0;
      G[n][1] = 0;

H[n][0] = 0;
      H[n][1] = 0;

}

//formula
    G[0][0] = x[0][0] + x[2][0];
  G[0][1] = x[0][1] + x[2][1];

G[1][0] = x[0][0] - x[2][0];
```

```
  G[1][1] = x[0][1] - x[2][1];

H[0][0] = x[1][0] + x[3][0];
  H[0][1] = x[1][1] + x[3][1];

H[1][0] = x[1][0] - x[3][0];
  H[1][1] = x[1][1] - x[3][1];

// Stage-2     e =
6.283185307179586 / N;

// X[k] = G[k] + WNnk H[k]
    k = 0;   t[0][0] = G[0][0] + (H[0][0] * cos (e * k) + H[0][1] *
sin (e * k));

t[0][1] = G[0][1] + (H[0][1] * cos (e * k) - H[0][0] * sin (e * k));

k = 1;
  t[1][0] = G[1][0] + (H[1][0] * cos (e * k) + H[1][1] * sin (e *
k));

t[1][1] = G[1][1] + (H[1][1] * cos (e * k) - H[1][0] * sin (e * k));

k = 2;   t[2][0] = G[0][0] + (H[0][0] * cos (e * k) + H[0][1] * sin
(e * k));

t[2][1] = G[0][1] + (H[0][1] * cos (e * k) - H[0][0] * sin (e * k));

k = 3;   t[3][0] = G[1][0] + (H[1][0] * cos (e * k) + H[1][1] * sin
(e * k));

t[3][1] = G[1][1] + (H[1][1] * cos (e * k) - H[1][0] * sin (e * k));

}


void
Point8 (int N, float x[8][2], float t[8][2])
{
   int a, b, c, d, i, j,
k; float e;

float X1[4][2], X2[4][2], G[4][2], H[4][2];
```

```
for (i = 0; i < 4; i++)

    {

X1[i][0] = x[2 * i][0];
      X1[i][1] = x[2 * i][1];

X2[i][0] = x[2 * i + 1][0];
      X2[i][1] = x[(2 * i) + 1][1];

}

Point4 (4, X1, G);                  //Decimation of Point8 signal to two Point4 signal.
  Point4 (4, X2, H);

// X[k] = G[k] + W H[k]     e =
6.283185307179586 / N;          // e= 2*(pie)/N   for
(k = 0; k < 4; k++)

    {
t[k][0] = G[k][0] + (H[k][0] * cos (e * k) + H[k][1] * sin (e * k));

t[k][1] = G[k][1] + (H[k][1] * cos (e * k) - H[k][0] * sin (e * k));

}

for (k = 0; k < 4; k++)

    {
d = k + 4;

t[d][0] = G[k][0] + (H[k][0] * cos (e * d) + H[k][1] * sin (e * d));

t[d][1] = G[k][1] + (H[k][1] * cos (e * d) - H[k][0] * sin (e * d));

}

}
```

OUTPUT



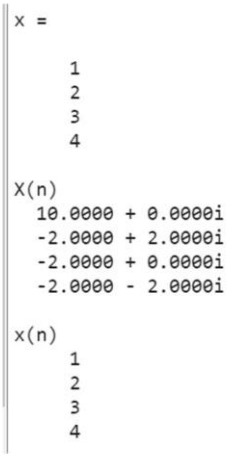
```
4pt or 8pt DITFFT = : 8
Enter the values of x[n]: 5 6 7 8 0 0 0 0


Input signal x[n] =  5.00  6.00  7.00  8.00  0.00  0.00  0.00  0.00


(FFT output)X[k] = :

 26.000 + j   0.000
  3.586 + j -16.899
 -2.000 + j   2.000
  6.414 + j  -2.899
 -2.000 + j   0.000
  6.414 + j   2.899
 -2.000 + j  -2.000
  3.586 + j  16.899
```

MATLAB

CODE

```
%program to find the fft
 x=[1;2;3;4]
y=fft(x);
disp('X(n)')
disp(y)



%calculate the ifft
 y_f=ifft(y);
disp('x(n)')
disp(y_f)
```

OUTPUT



```
x =

     1
     2
     3
     4

X(n)
  10.0000 + 0.0000i
  -2.0000 + 2.0000i
  -2.0000 + 0.0000i
  -2.0000 - 2.0000i

x(n)
     1
     2
     3
     4
```

CONCLUSION

Got introduced to the concept of Fast Fourier Transform which is converting a signal in time domain to a signal in frequency domain. Was able to learn its properties and implement the same in C language and MATLAB.