**DIGITAL SIGNAL PROCESSING LAB EXPT.7**

**NIKHIL ROUT**

**22BEC1020**

**AIM: PERFORMING DFT AND IDFT OF AN 8-POINT DISCRETE SIGNAL**

**C CODE**

#include <math.h>  
#include <stdio.h>  
  
  
void calculateDFT()  
{  
int xn[8];  
float Xr[8];  
float Xi[8];  
int i, k, n, N = 0;  
  
for (i = 0; i < 8; i++) {  
  
printf("Enter the value "  
"of x[%d]: ",  
i);  
scanf("%d", &xn[i]);  
}  
  
printf("Enter the number of "  
"points in the DFT: ");  
scanf("%d", &N);  
for (k = 0; k < N; k++) {  
Xr[k] = 0;  
Xi[k] = 0;  
for (n = 0; n < 8; n++) {  
Xr[k]  
= (Xr[k]  
+ xn[n] \* cos(2 \* 3.141592 \* k \* n / N));  
Xi[k]  
= (Xi[k]  
- xn[n] \* sin(2 \* 3.141592 \* k \* n / N));  
}  
  
printf("(%f) + j(%f)\n", Xr[k], Xi[k]);  
}  
}  
  
void calculateIDFT(){  
   int xn[8];  
float Xr[8];  
float Xi[8];  
int i, k, n, N = 0;  
  
for (i = 0; i < 8; i++) {  
  
printf("Enter the value "  
"of x[%d]: ",  
i);  
scanf("%d", &xn[i]);  
}  
  
printf("Enter the number of "  
"points in the IDFT: ");  
scanf("%d", &N);  
for (k = 0; k < N; k++) {  
Xr[k] = 0;  
Xi[k] = 0;  
for (n = 0; n < 8; n++) {  
Xr[k]  
= (Xr[k]  
+ xn[n] \* cos(2 \* 3.141592 \* k \* n / N));  
Xi[k]  
= (Xi[k]  
+ xn[n] \* sin(2 \* 3.141592 \* k \* n / N));  
}  
xn[n] = xn[n] / N;  
printf("(%f) + j(%f)\n", Xr[k], Xi[k]);  
}  
}  
  
  
int main()  
{  
  
calculateDFT();  
calculateIDFT();  
  
return 0;  
}

**CCS CONSOLE OUTPUT**



