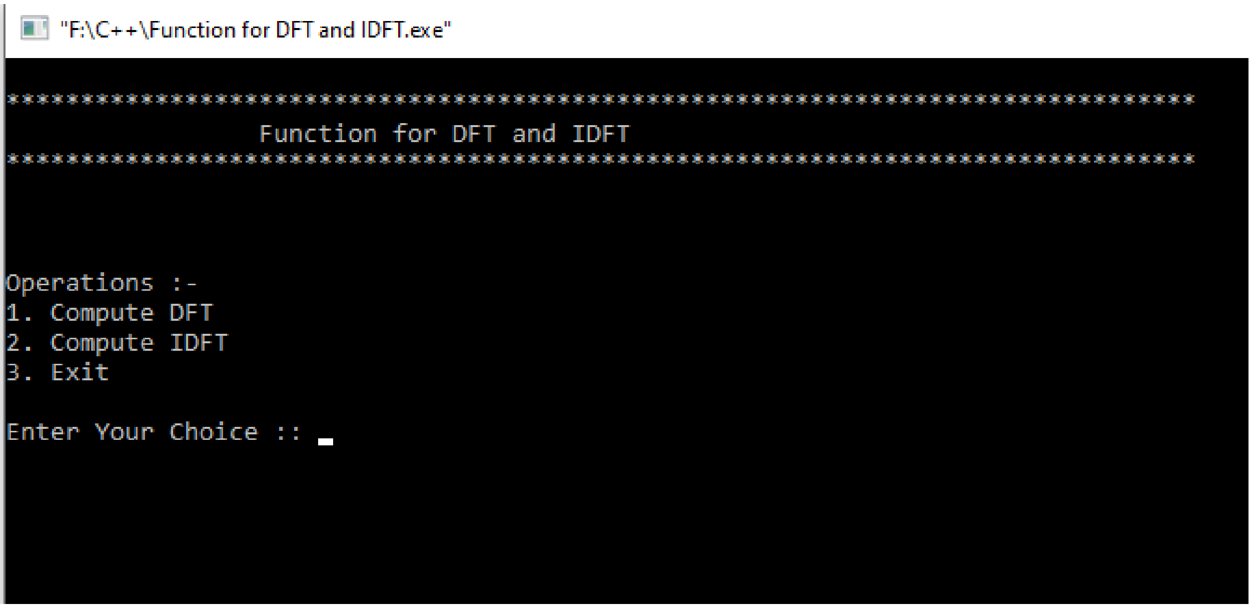
1. Create Function for DFT and IDFT and test:

Code:

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

#define PI 3.1415
float static input[100],ip_real[100],ip_img[100];

int main(void)
{
   int k,n,N,c;
   float M;

printf("\n*******************************************************************************\n");
   printf("\t\t Function for DFT and IDFT\n");
printf("*******************************************************************************\n\n");

   do
   {
      printf("\n\nOperations :-\n");
      printf("1. Compute DFT \n");
      printf("2. Compute IDFT \n");
      printf("3. Exit \n");
      printf("\nEnter Your Choice :: ");
      scanf("%d",&c);
      switch(c)
      {
      case 1:
         printf("\n");
         printf("Enter the numbers of samples in the sequence N = ");
         scanf("%d",&N);     // Number of sample N
         printf("\nEnter the of samples of sequence X(n)\n");

         for(n = 0 ; n < N; n++)
         {
            printf("X(%d) = ",n);
            scanf("%f",&input[n]);
         }

         for(k = 0 ; k < N ; k++)
         {
            ip_img[k] = ip_real[k] = 0.0;
            for(n = 0 ; n < N ; n++)
            {
               ip_real[k] = ip_real[k] + input[n]*cos((2*PI*k*(n-N))/N);
               ip_img[k] = ip_img[k] + input[n]*sin((2*PI*k*(n-N))/N);
            }
            ip_img[k] = ip_img[k]*(-1.0);
         }
         printf("The DFT for the Sequence :-\n");
         for(k=0;k<N;k++)
         {
            printf("\nX(%d) = %.1f \t\t %.1fj",k,ip_real[k],ip_img[k]);
         }
         break;
      case 2:
         printf("\n");
         printf("Enter the numbers of samples in the sequence N = ");
         scanf("%f",&M);     // Number of sample N
         printf("\nEnter the of samples of sequence\n X(n) as 0.0  j0.0\n");

         for(n = 0 ; n < M; n++)
         {
            printf("X(%d) = ",n);
            scanf("%f%f",&ip_real[n],&ip_img[n]);
         }

         for(k = 0 ; k < M ; k++)
         {
            //ip_img[k] = ip_real[k] = 0.0;
            input[k] = 0;
            for(n = 0 ; n < M ; n++)
            {
               input[k] = input[k]+ip_real[n]*cos((2*PI*k*n)/M)-ip_img[n]*sin((2*PI*k*n)/M);
            }
            input[k] = input[k]/M;
         }
         printf("\n\nThe IDFT for the Sequence :-\n");
         for(k=0;k<M;k++)
         {
            printf("\nX(%d) = %.2f",k,input[k]);
         }

         break;

      default:
```

```
printf("************************************************************************
********\n\n");
          printf("you entered Invalid choice... Try again !\n");

       }
   }while(c!=3);

}
```

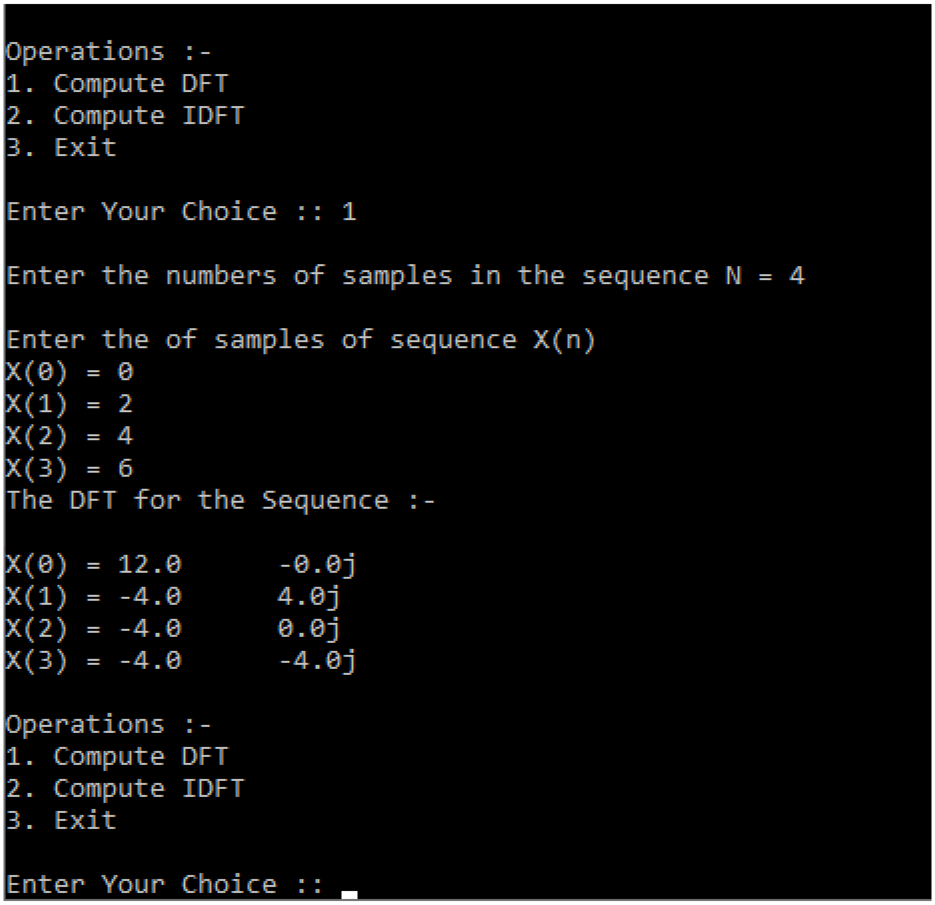
Output:-



"F:\C++\Function for DFT and IDFT.exe"

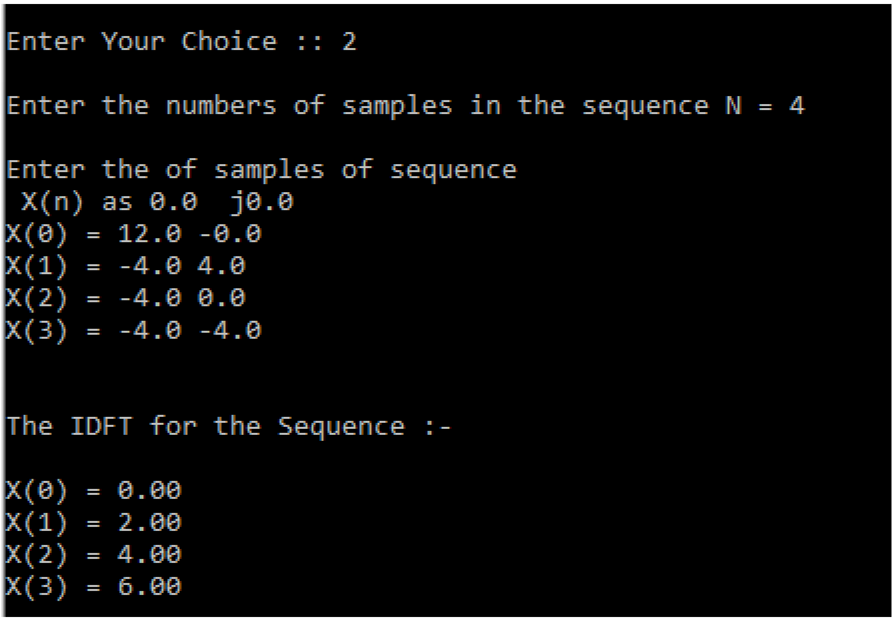
```
********************************************************************************
                 Function for DFT and IDFT
********************************************************************************



Operations :-
1. Compute DFT
2. Compute IDFT
3. Exit

Enter Your Choice :: _
```

Compute DFT :

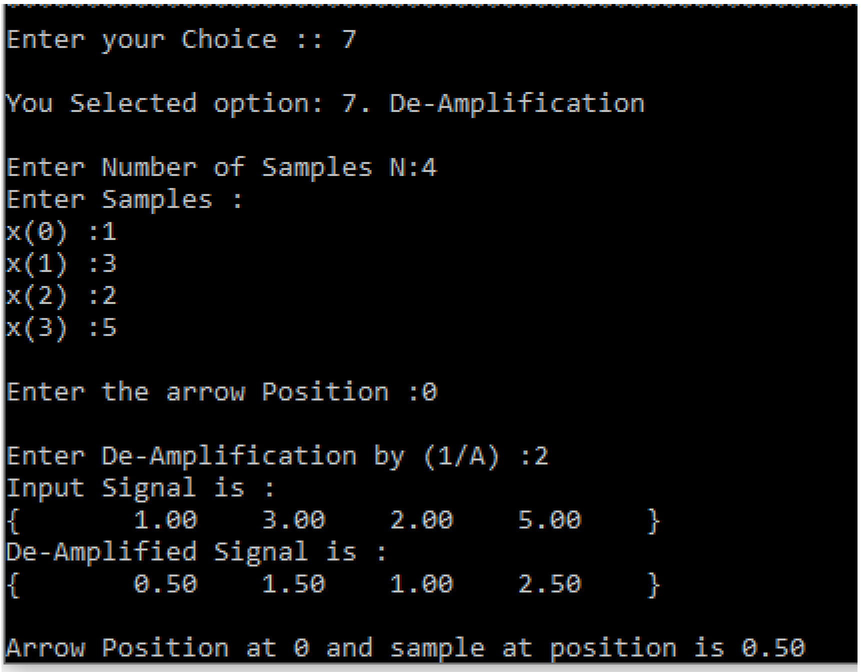


```
Operations :-
1. Compute DFT
2. Compute IDFT
3. Exit

Enter Your Choice :: 1

Enter the numbers of samples in the sequence N = 4

Enter the of samples of sequence X(n)
X(0) = 0
X(1) = 2
X(2) = 4
X(3) = 6
The DFT for the Sequence :-

X(0) = 12.0      -0.0j
X(1) = -4.0      4.0j
X(2) = -4.0      0.0j
X(3) = -4.0      -4.0j

Operations :-
1. Compute DFT
2. Compute IDFT
3. Exit

Enter Your Choice :: _
```

Compute IDFT :



```
Enter Your Choice :: 2

Enter the numbers of samples in the sequence N = 4

Enter the of samples of sequence
 X(n) as 0.0  j0.0
X(0) = 12.0 -0.0
X(1) = -4.0 4.0
X(2) = -4.0 0.0
X(3) = -4.0 -4.0


The IDFT for the Sequence :-

X(0) = 0.00
X(1) = 2.00
X(2) = 4.00
X(3) = 6.00
```



```
Enter your Choice :: 7

You Selected option: 7. De-Amplification

Enter Number of Samples N:4
Enter Samples :
x(0) :1
x(1) :3
x(2) :2
x(3) :5

Enter the arrow Position :0

Enter De-Amplification by (1/A) :2
Input Signal is :
{       1.00    3.00    2.00    5.00    }
De-Amplified Signal is :
{       0.50    1.50    1.00    2.50    }

Arrow Position at 0 and sample at position is 0.50
```