

## FFT

```
#include <iostream>
```

```
#include <math.h>
```

```
#include <complex>
```

```
using namespace std;
```

```
const double PI = acos(-1);
```

```
typedef complex<double> cd;
```

```
cd *FFT(cd S[], int n)
```

```
{
```

```
    cd *A = new cd[n];
```

```
    if (n <= 1)
```

```
    {
```

```
        A[0] = S[0];
```

```
        return A;
```

```
    }
```

```
    cd *X = new cd[n / 2];
```

```
    cd *Y = new cd[n / 2];
```

```
    for (int i = 0; i < n / 2; i++)
```

```
    {
```

```
        X[i] = S[2 * i];
```

```
        Y[i] = S[2 * i + 1];
```

```
    }
```

```
    cd *B = FFT(X, n / 2);
```

```
    cd *C = FFT(Y, n / 2);
```

```
    double power = -2.0 * PI / n;
```

```
    cd w(cos(power), sin(power));
```

```
    cd p = 1.0;
```

```

for (int k = 0; k < n / 2; k++)
{
    cd term = p * C[k];
    A[k] = B[k] + term;
    A[k + n / 2] = B[k] - term;
    p *= w;
}
delete[] X;
delete[] Y;
delete[] B;
delete[] C;
return A;
}

int main()
{
    int n;
    printf("Enter the number of elements in the sequence: ");
    scanf("%d", &n);
    cd x[n];
    printf("Enter the sequence:\n");
    for (int i = 0; i < n; i++)
    {
        double real, imag;
        printf("Enter real and imag value for x^%d: ", i);
        scanf("%lf %lf", &real, &imag);
        x[i] = {real, imag};
    }
    cd *A = new cd[n];

```

```
A = FFT(x, n);  
printf("FFT Result:\n");  
for (int i = 0; i < n; ++i)  
{  
    printf("(%.0lf + %.0lfi)\n", A[i].real(), A[i].imag());  
}  
return 0;  
}
```

## OUTPUT

```
Enter the number of elements in the sequence: 4  
Enter the sequence:  
Enter real and imag value for x^0: 1.0  
2.0  
Enter real and imag value for x^1: 3.0  
4.0  
Enter real and imag value for x^2: 5.0  
6.0  
Enter real and imag value for x^3: 7.0  
8.0  
FFT Result:  
(16 + 20i)  
(-8 + -0i)  
(-4 + -4i)  
(0 + -8i)
```