```
/*************************************************************************
* Program is for CMPE 245 class use, see Dr. Harry Li's lecture notes for details *
* Reference: Digital Signal Processing, by A.V. Oppenhaim;                        *
* fft.c for calculting 4 points input, but you can easily expand this to 2^x inputs; *
* Version: x0.1;            Date: Sept. 2009;                                     *
**************************************************************************/
#include <stdio.h>
#include <math.h>
struct Complex
{       double a;       //Real Part
        double b;       //Imaginary Part
}       X[5], U, W, T, Tmp;

void FFT(void)
{
        int M = 2;
        int N = pow(2, M);

        int i = 1, j = 1, k = 1;
        int LE = 0, LE1 = 0;
        int IP = 0;

        for (k = 1; k <= M; k++)
        {
                LE = pow(2, M + 1 - k);
                LE1 = LE / 2;

                U.a = 1.0;
                U.b = 0.0;

                W.a = cos(M_PI / (double)LE1);
                W.b = -sin(M_PI/ (double)LE1);

                for (j = 1; j <= LE1; j++)
                {
                        for (i = j; i <= N; i = i + LE)
                        {
                                IP = i + LE1;
                                T.a = X[i].a + X[IP].a;
                                T.b = X[i].b + X[IP].b;
                                Tmp.a = X[i].a - X[IP].a;
                                Tmp.b = X[i].b - X[IP].b;
                                X[IP].a = (Tmp.a * U.a) - (Tmp.b * U.b);
                                X[IP].b = (Tmp.a * U.b) + (Tmp.b * U.a);
                                X[i].a = T.a;
                                X[i].b = T.b;
```

```
            }
            Tmp.a = (U.a * W.a) - (U.b * W.b);
            Tmp.b = (U.a * W.b) + (U.b * W.a);
            U.a = Tmp.a;
            U.b = Tmp.b;
        }
    }

    int NV2 = N / 2;
    int NM1 = N - 1;
    int K = 0;

    j = 1;
    for (i = 1; i <= NM1; i++)
    {
        if (i >= j) goto TAG25;
        T.a = X[j].a;
        T.b = X[j].b;

        X[j].a = X[i].a;
        X[j].b = X[i].b;
        X[i].a = T.a;
        X[i].b = T.b;
TAG25:  K = NV2;
TAG26:  if (K >= j) goto TAG30;
        j = j - K;
        K = K / 2;
        goto TAG26;
TAG30:  j = j + K;
    }
}

int main(void)
{
    float arr[5] = {0.0, 2.0, 3.0, 4.0, 4.0};
    int i;
    for (i = 0; i < 5; i++)
    {
        X[i].a = arr[i];
        X[i].b = 0.0;
    }

    printf ("*********Before*********\n");
    for (i = 1; i <= 4; i++)
        printf ("X[%d]:real == %f  imaginary == %f\n", i, X[i].a, X[i].b);
    FFT();
    printf ("\n\n**********After*********\n");
    for (i = 1; i <= 4; i++)
        printf ("X[%d]:real == %f  imaginary == %f\n", i, X[i].a, X[i].b);

    return 0;
}
```