```c
/* Code originally taken from the following URL:
     http://svn.arhuaco.org/svn/src/emqbit/tools/emqbit-bench/
*/

/*
 * Authors:
 *    Jorge Victorino
 *    Andres Calderon   andres.calderon@emqbit.com
 *
 * This program is free software; you can redistribute it and/or modify it
 * under the terms of the GNU General Public License as published by the
 * Free Software Foundation; either version 2 of the License, or (at your
 * option) any later version.
 */


#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "cfft.h"
#include "common.h"

complex *tableW;
int *bndx;
int *ndx;

void fft_init (int N)
{
  int i, j;

  tableW = malloc ((N / 2) * sizeof (complex));
  bndx = malloc (N * sizeof (int));
  ndx = malloc ((N / 2) * sizeof (int));

  ndx[0] = 0;
  for (i = 1; i < N / 2; i = i * 2)
  {
    for (j = 0; j < i; j++)
    {
      ndx[j] *= 2;
      ndx[j + i] = ndx[j] + 1;
    }
  }

  bndx[0] = 0;
  for (i = 1; i < N; i = i * 2)
  {
    for (j = 0; j < i; j++)
    {
      bndx[j] *= 2;
      bndx[j + i] = bndx[j] + 1;
    }
  }

  for (i = 0; i < N / 2; i++)
  {
    tableW[i].r = cos (ndx[i] * 2.0F * M_PI / (float) N);
    tableW[i].i = -sin (ndx[i] * 2.0F * M_PI / (float) N);
  }
}

void fft_end ()
{
  free (ndx);
  free (bndx);
  free (tableW);
}

void fft_exec (int N, complex * in)
{
  unsigned int n = N;
  unsigned int a, b, i, j, k, r, s;
  complex w, p;

  for (i = 1; i < N; i = i * 2)
  {
    n = n >> 1;
    for (k = 0; k < i; k++)
    {
      w = tableW[k];

      r = 2 * n * k;
      s = n * (1 + 2 * k);

      for (j = 0; j < n; j++)
      {
        a = j + r;
        b = j + s;
        cmult (p, w, in[b]);      //6 flop
        csub (in[b], in[a], p);   //2 flop
        cadd (in[a], in[a], p);   //2 flop
      }
    }
  }
}

```