```c
# include <stdlib.h>
# include <stdio.h>
# include <math.h>
# include <time.h>

# include "cordic.h"

/****************************************************************************/

double compute_angle ( int idx, int nIter )

/****************************************************************************/
/*
   Purpose:

      Angle Array in Binary Tree Representation

   Discussion:


   Licensing:

      This code is distributed under the GNU LGPL license.

   Modified:

      2012.04.16

   Author:

      Young Won Lim

   Parameters:

*/
{
  double angle = 0.0;
  int i, j;
  char s[32];

  for (i=0; i<nIter; i++) {
    j = 1 << i;
    if (idx & (1 << i)) {
      angle += atan( 1. / j );
      s[nIter-i-1] = '1';
    } else {
      angle -= atan( 1. / j );
      s[nIter-i-1] = '0';
    }
    printf("i=%d j=%d 1/j=%f atan(1/j)=%f \n", i, j, 1./j, atan(1./j)*180/3.1416);
  }
  s[nIter] = '\0';

  printf("%d %d %s ---> %f \n", nIter, idx, s, angle*180/3.1416);



  return angle;
}


int compare (const void *a , const void *b) {
  return ( *(double *) a - *(double *) b );
}


int main () {

  int nIter = 4;
  int nAngle = 1 << nIter;
```

```c
    int i;
    double A[nAngle];

    for (i=0; i<nAngle; ++i) {

        A[i] = compute_angle(i, nIter);
    }

    for (i=0; i<nAngle; i++) {
        printf("A[%d] = %f \n", i, A[i]);
    }


    return 0;

}
```