```
runner% cat dice2.c
/* dice: Gather statistics about rolling
 * two dice.  A Monte Carlo simulation
 * using the random number generator
 * drand48() with initializer srand48(int),
 * both in <stdlib.h>
 */
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define MAXR 1000000

void initarray(int d[]);
void printarray(int d[]);
void printstar(int n);
void fancyprint(int d[]);
void roll(int *d1p, int *d2p);

void main(void)
{
   int d[13]; /* 2-12 hold num of rolls */
   int d1, d2; /* results of two rolls */
   int i;
   int starttime = time(NULL);
   initarray(d);
   srand48((long)starttime);
   printf("Total rolls: %1d\n\n", MAXR);
   for (i = 0; i < MAXR; i++) {
      roll(&d1, &d2);
      d[d1 + d2]++;
   }
   printarray(d);
   fancyprint(d);
   printf("Elapsed time: %ld seconds\n",
   time(NULL) - starttime);
   exit(0);
}

/* initarray: initialize statistic array */
void initarray(int d[])
{
   int i;
   for (i = 2; i < 13; i++)
      d[i] = 0;
}

/* printarray: print each num of rolls */
void printarray(int d[])
{
   int i;
   double e[13] = {0, 0,
      1.0/36.0, 2.0/36.0, 3.0/36.0, 4.0/36.0,
      5.0/36.0, 6.0/36.0, 5.0/36.0, 4.0/36.0,
      3.0/36.0, 2.0/36.0, 1.0/36.0};
   printf("Sum  Times   Frequency");
   printf("      Exact     Diff\n\n");
   for (i = 2; i < 13; i++)
      printf("%2d %7d %11.7f %10.7f %8.4f\n",
        i,d[i],(double)d[i]/MAXR*100.0,
        e[i]*100.0, ((double)(d[i]) -
        e[i]*MAXR)/MAXR*100.0);
}

/* printstar: print n stars */
void printstar(int n)
{
   while (n > 0) {
      printf("*");
      n--;
   }
}

/* fancyprint: print bar graph */
void fancyprint(int d[])
{
   int i;
   printf("\n");
   for (i = 2; i < 13; i++) {
      printf("Sum:%3d |", i);
      printstar(300*d[i]/MAXR);
      printf("\n");
   }
   printf("\n");
}

/* roll: simulate rolling two dice */
void roll(int *d1p, int *d2p)
{
   *d1p = (int) (6.0*drand48() + 1.0);
   *d2p = (int) (6.0*drand48() + 1.0);
}

runner% cc -o dice2 dice2.c
runner% dice2
Total rolls: 1000000

Sum  Times   Frequency      Exact     Diff

 2   27701   2.7701000  2.7777778  -0.0077
 3   55911   5.5911000  5.5555556   0.0355
 4   83225   8.3225000  8.3333333  -0.0108
 5  111627  11.1627000 11.1111111   0.0516
 6  138101  13.8101000 13.8888889  -0.0788
 7  166523  16.6523000 16.6666667  -0.0144
 8  139185  13.9185000 13.8888889   0.0296
 9  111106  11.1106000 11.1111111  -0.0005
10   83167   8.3167000  8.3333333  -0.0166
11   55444   5.5444000  5.5555556  -0.0112
12   28010   2.8010000  2.7777778   0.0232

Sum:  2 |*****
Sum:  3 |***********
Sum:  4 |****************
Sum:  5 |**********************
Sum:  6 |****************************
Sum:  7 |**********************************
Sum:  8 |****************************
Sum:  9 |**********************
Sum: 10 |****************
Sum: 11 |***********
Sum: 12 |*****

Elapsed time: 16 seconds
runner%
```