```
runner% cat tree.c
/* Print a tree
 * Written by NR Wagner, Sept 25, 1996
 */
#include <stdio.h>

void chars(int, char);

void main(void)
{
   int n;
   int row;
   /* read number of rows for top of tree */
   scanf("%i", &n);
   /* print the main part */
   for (row = 1; row <= n; row++) {
      chars(n - row, ' ');  /* initial blanks */
      chars(row*2 - 1,'*'); /* stars for tree */
      chars(1, '\n');       /* next line */
   }
   /* print the tree trunk */
   if (n >= 3)
      for (row = 1; row <= 3; row++) {
         chars(n - 2, ' ');
         chars(3, '*');
         chars(1, '\n');
      }
   /* print the base of the tree */
   if (n >= 4) {
      chars(n - 4, ' ');
      chars(7, '*');
      chars(1, '\n');
   }
}

/* chars: print n of an input character */
void chars(int n, char ch)
{
  int i;
   for (i = 0; i < n; i++)
      printf("%c", ch);
}

runner% cc -o tree tree.c
runner% tree
12
```

```
           *
          ***
         *****
        *******
       *********
      ***********
     *************
    ***************
   *****************
  *******************
 *********************
***********************
          ***
          ***
          ***
        *******
runner% tree
2
 *
***
runner% tree
3
  *
 ***
*****
 ***
 ***
 ***
runner% tree
4
   *
  ***
 *****
*******
  ***
  ***
  ***
*******
runner% tree
8
       *
      ***
     *****
    *******
   *********
  ***********
 *************
***************
      ***
      ***
      ***
    *******
```

```
runner% cat graph.c
/* Bar graph of a function. */
/* Neal R. Wagner, Oct 9, 1996 */
#include <stdio.h>
#include <math.h>

double f(double);
void chars(int, char);

void main(void)
{
   double x, y; /* x and y coords on graph */
   double a, b; /* let x go from a to b */
   double h;    /* in steps of h */
   scanf("%lf %lf %lf", &a, &b, &h);
   for (x = a; x <= b; x = x + h) {
      printf("%5.2f|", x);
      y = f(x);
      chars((int)(2.0/h)*y, '*');
      printf("\n");
   }
}

/* f: the function to graph */
double f(double x)
{
   return sin(x);
}

/* chars: print character ch n times */
void chars(int n, char ch)
{
   int i;
   for (i = 0; i < n; i++)
      printf("%c", ch);
}
runner% lint -m -u graph.c -lm
function returns value which is always ignored
    printf              scanf
runner% cc -o graph graph.c -lm
```

```
runner% ./graph
0.0  3.14159  0.1
 0.00|
 0.10|*
 0.20|***
 0.30|*****
 0.40|*******
 0.50|*********
 0.60|***********
 0.70|************
 0.80|**************
 0.90|***************
 1.00|****************
 1.10|*****************
 1.20|******************
 1.30|*******************
 1.40|*******************
 1.50|*******************
 1.60|*******************
 1.70|*******************
 1.80|*******************
 1.90|******************
 2.00|******************
 2.10|*****************
 2.20|****************
 2.30|**************
 2.40|*************
 2.50|***********
 2.60|**********
 2.70|********
 2.80|******
 2.90|****
 3.00|**
 3.10|
Next, use double f(double x){ return 2 - x*x;}
runner% ./graph
0.0  1.414  0.1
 0.00|****************************************
 0.10|***************************************
 0.20|***************************************
 0.30|**************************************
 0.40|************************************
 0.50|***********************************
 0.60|********************************
 0.70|******************************
 0.80|***************************
 0.90|***********************
 1.00|********************
 1.10|***************
 1.20|***********
 1.30|******
 1.40|
```