

```

//
// bowlingpins.c
//
// Blake Hall
// Math 442
// Jim Fix
// Fall 2013
//

#include <stdio.h>

int main() {
    // Initialize variable
    int numRows;
    int counter;
    int i;
    int j;

    // Ask user for number of rows
    printf("How many rows would you like? ");
    scanf("%d",&numRows);
    counter = numRows;

    // Print out a picture of the bowling pins
    while (counter != 0) {
        for (i = 1; i <= counter; i++) {
            printf(" ");
        }
        for (j = 1; j <= (numRows - counter + 1); j++) {
            printf("* ");
        }
        printf("\n");
        counter--;
    }
}

```

```

//
// highandlow.c
//
// Blake Hall
// Math 442
// Jim Fix
// Fall 2013
//
// Picks a random number, then asks the user to guess that number.
// If guess is too high, it asks for a lower number; if too low,
// it asks for a higher number. Continues until correct number is
// guessed.
//
//

#include <stdio.h>
#include <time.h>

// Generate a random number.
int generateNum() {
    srand(time(NULL));
    int number = rand() % 100 + 1;
    return number;
}

// Plays the game!
int main() {

    // Initialize important values.

```

```

int guess;
int gameNum = generateNum();

// Get guess from user.
printf("What is your guess?: ");
scanf("%d", &guess);

// Simple if and while loops that ask the user for guesses until
// he is correct.
if (guess == gameNum) {
    printf("You're right!\n");
    return;
} else {
    while (guess != gameNum) {
        if (guess <= gameNum) {
            printf("Higher! Try again!\n");
        } else {
            printf("Lower! Try again!\n");
        }
        printf("What is your next guess? ");
        scanf("%d",&guess);
    }
    printf("Yup! You got it!\n");
    return;
}
}

```

```

//
// singrapher.c
//
// Blake Hall
// Math 442
// Jim Fix
// Fall 2013
//

```

```

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

```

```

// This function takes the power of a given value and returns it
double powerOf(double x, int n) {
    double pow = x;
    while (n > 1) {
        pow = pow*x;
        n--;
    }
    return pow;
}

```

```

// This function determines the value of the power of sin(x), determines
// where it should place the *, then prints out a single slice of the
// graph
double printSlice(double degree, int power) {
    // Find value of (sin(x))^n, and scale by 10
    double value = powerOf(sin(degree),power)*10;
    // Define our interval
    double i = -10.0;
    double j = -9.0;

    // Iterate through ~20 spaces to determine if spot should be
    // a space, a line, or a star.
    while (j <= 10) {
        // If "j" (right side of interval) is 1, we want to draw the axis line
    }
}

```

```

    if (j == 1.0) {
        printf("|");
        i++;
        j++;
    } else {
        // If our value falls within our interval, print a star
        if (i <= value && value <= j) {
            printf("*");
            i++;
            j++;
            // Otherwise, print a space
        } else {
            printf(" ");
            i++;
            j++;
        }
    }
}
printf("\n");

/* USING ROUND (disabled)
// Determine location of * by rounding value, taking the
// absolute value, then casting it as an int. Note that
// starLoc is positive. It's location if value is negative
// is taken care of in the next section.
int starLoc = abs((int)round(value*10));
int i;

// If value is less than 0, print the appropriate half of the slice
// of the graph.
if (value < 0.0) {
    for (i = 0; i < (10 - starLoc); i++) {
        printf(" ");
    }
    printf("*");
    for (i = 0; i < (starLoc); i++) {
        printf(" ");
    }
    printf("|          \n");
} else {
    // Otherwise, print a blank half then print out the appropriate
    // second half.
    printf("          |");
    for(i = 0; i < (starLoc - 1); i++) {
        printf(" ");
    }
    printf("*");
    for(i = 0; i < (10 - starLoc); i++) {
        printf(" ");
    }
    printf("\n");
}*/
}

// Runs the application
int main() {
    int power;
    double degree = -3.14;
    printf("Enter the power you wish to raise sin(x) to: ");
    scanf("%d",&power);
    // Will use printSlice over the given range of degrees.
    while (degree <= 3.452) {
        printSlice(degree,power);
        degree = degree + 0.314;
    }
}

```

```
    return;  
}
```