

2. Modify Programming Project 5 from Chapter 5 so that it uses a function to compute the amount of income tax. When passed an amount of taxable income, the function will return the tax due.
3. Modify Programming Project 9 from Chapter 8 so that it includes the following functions:


```
void generate_random_walk(char walk[10][10]);
void print_array(char walk[10][10]);
```

main first calls `generate_random_walk`, which initializes the array to contain ' .' characters and then replaces some of these characters by the letters A through Z, as described in the original project. main then calls `print_array` to display the array on the screen.
4. Modify Programming Project 16 from Chapter 8 so that it includes the following functions:


```
void read_word(int counts[26]);
bool equal_array(int counts1[26], int counts2[26]);
```

main will call `read_word` twice, once for each of the two words entered by the user. As it reads a word, `read_word` will use the letters in the word to update the `counts` array, as described in the original project. (main will declare two arrays, one for each word. These arrays are used to track how many times each letter occurs in the words.) main will then call `equal_array`, passing it the two arrays. `equal_array` will return `true` if the elements in the two arrays are identical (indicating that the words are anagrams) and `false` otherwise.
5. Modify Programming Project 17 from Chapter 8 so that it includes the following functions:


```
void create_magic_square(int n, char magic_square[n][n]);
void print_magic_square(int n, char magic_square[n][n]);
```

After obtaining the number n from the user, main will call `create_magic_square`, passing it an $n \times n$ array that is declared inside main. `create_magic_square` will fill the array with the numbers 1, 2, ..., n^2 as described in the original project. main will then call `print_magic_square`, which will display the array in the format described in the original project. *Note:* If your compiler doesn't support variable-length arrays, declare the array in main to be 99×99 instead of $n \times n$ and use the following prototypes instead:

```
void create_magic_square(int n, char magic_square[99][99]);
void print_magic_square(int n, char magic_square[99][99]);
```
6. Write a function that computes the value of the following polynomial:

$$3x^5 + 2x^4 - 5x^3 - x^2 + 7x - 6$$

Write a program that asks the user to enter a value for x , calls the function to compute the value of the polynomial, and then displays the value returned by the function.
7. The power function of Section 9.6 can be made faster by having it calculate x^n in a different way. We first notice that if n is a power of 2, then x^n can be computed by squaring. For example, x^4 is the square of x^2 , so x^4 can be computed using only two multiplications instead of three. As it happens, this technique can be used even when n is not a power of 2. If n is even, we use the formula $x^n = (x^{n/2})^2$. If n is odd, then $x^n = x \times x^{n-1}$. Write a recursive function that computes x^n . (The recursion ends when $n = 0$, in which case the function returns 1.) To test your function, write a program that asks the user to enter values for x and n , calls `power` to compute x^n , and then displays the value returned by the function.
8. Write a program that simulates the game of craps, which is played with two dice. On the first roll, the player wins if the sum of the dice is 7 or 11. The player loses if the sum is 2, 3,