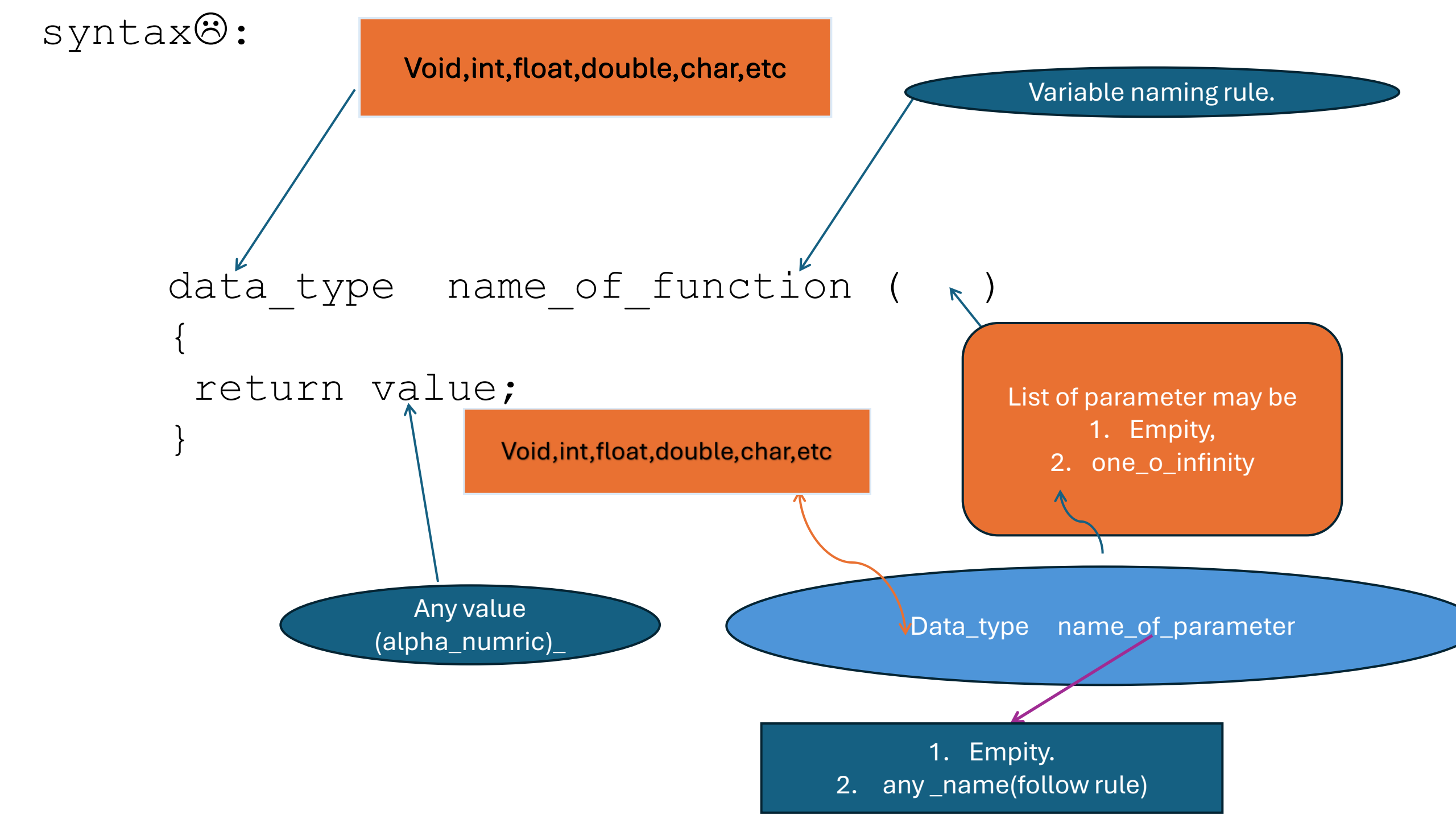
function☹

- Pehlay concept clare hay to function koi alag shay nahi hay.
- Function ham is liya use kartay hain jis k through ham code ko proper order and tarteeb ma rakhsakay taki baad ma parnay ya smaj nay k liy asani ho function k baghere b ham wo sara kam karsaktay hain magar. For example sum,avg,minus,double,print,calculate; ye sub ham wesay b karchukay hain.
- Real example:
- Kitchen(ma ham saray cheez to nahi rakhsaktayna jaysay ki goat,cow,bike,car,books,hous,store material etc. So, in sub ko apni apni jaga rakhnay k liya function use kartay hain. Umeed karta hun ki samahj ma aya hoga.

syntax☹:

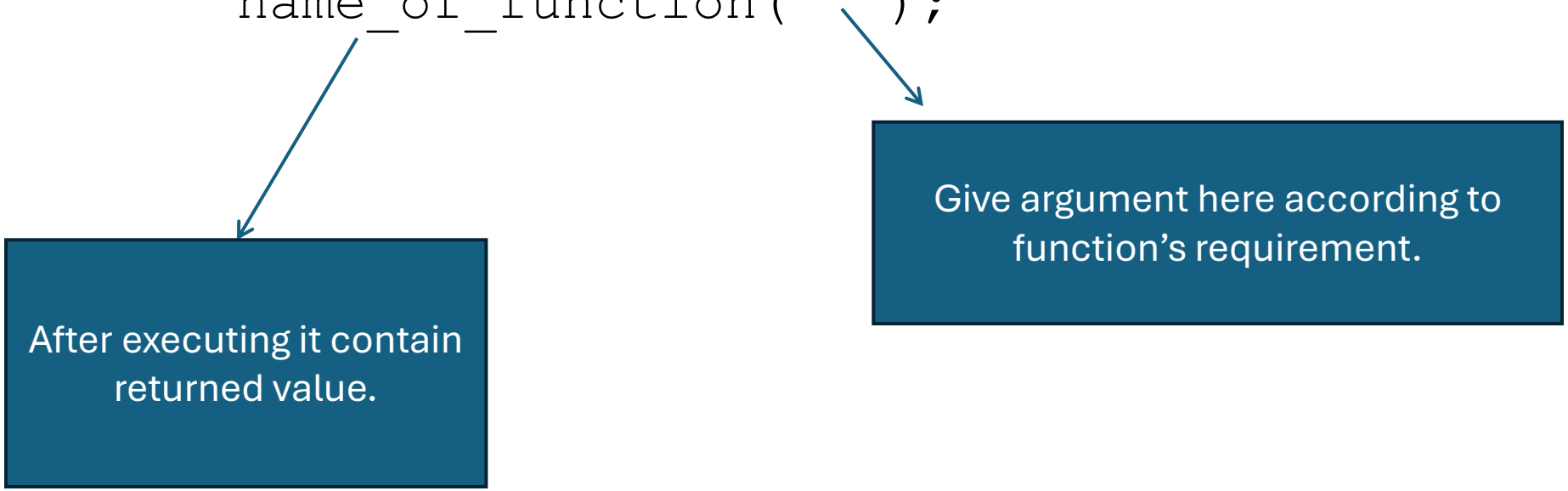

Void,int,float,double,char,etc
Variable naming rule.
data_type name_of_function ( )
{
return value;
}
List of parameter may be
1. Empity,
2. one_o_infinity
Void,int,float,double,char,etc
Any value
(alpha_numric)_
Data_type name_of_parameter
1. Empity.
2. any _name(follow rule)

Calling a function

To use that block of code that we separate for use when need.

syntax:

```
name_of_function(   );
```



To use returned value

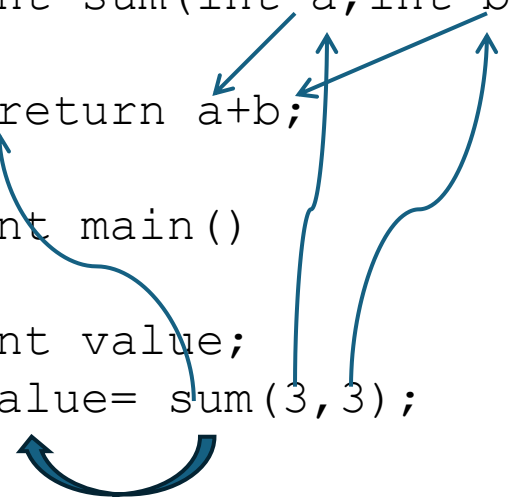
1. We use assignment operator to assign any other variable
2. Or direct use where need.

Syntax for using returned value.

If function data_type is int
Then make integer variable to assign returned value,
If direct use according to situation.

For example

```
#include<stdoio.h>
int sum(int a,int b)
{
 return a+b;
}
int main()
{
int value;
Value= sum(3,3);
}
```



EXCERCISE

1. make a c program using a function for sum and average two integer.

2. make a c program to print table of 5 up to ten like

5 x 1 = 5
5 x 2 = 10
.
.

By using function
Input 5 → direct print in function
using loop i=1 →i=10; format table

3. make a c program using function for ATM machine

- function return amount
- Input balance.
- Return amount is equal to old amount $1000 add with current balance as input of funciton.
- Print returned amount.on terminal .

Exercises

1. The following function, which computes the area of a triangle, contains two errors. Locate the errors and show how to fix them. (*Hint:* There are no errors in the formula.)

```
double triangle_area(double base, height)
double product;
{
  product = base * height;
  return product / 2;
}
```

2. Write a function `check(x, y, n)` that returns 1 if both `x` and `y` fall between 0 and `n` – 1, inclusive. The function should return 0 otherwise. Assume that `x`, `y`, and `n` are all of type `int`.

3. Write a function `gcd(m, n)` that calculates the greatest common divisor of the integers `m` and `n`. (Programming Project 2 in Chapter 6 describes Euclid's algorithm for computing the GCD.)

4. Write a function `day_of_year(month, day, year)` that returns the day of the year (an integer between 1 and 366) specified by the three arguments.

5. Write a function `num_digits(n)` that returns the number of digits in `n` (a positive integer). *Hint:* To determine the number of digits in a number *n*, divide it by 10 repeatedly. When *n* reaches 0, the number of divisions indicates how many digits *n* originally had.

6. Write a function `digit(n, k)` that returns the k^{th} digit (from the right) in `n` (a positive integer). For example, `digit(829, 1)` returns 9, `digit(829, 2)` returns 2, and `digit(829, 3)` returns 8. If `k` is greater than the number of digits in `n`, have the function return 0.

7. Suppose that the function `f` has the following definition:

   `int f(int a, int b) { ... }`

   Which of the following statements are legal? (Assume that `i` has type `int` and `x` has type `double`.)

   (a) `i = f(83, 12);`
   (b) `x = f(83, 12);`
   (c) `i = f(3.15, 9.28);`
   (d) `x = f(3.15, 9.28);`
   (e) `f(83, 12);`

8. Which of the following would be valid prototypes for a function that returns nothing and has one `double` parameter?

   (a) `void f(double x);`
   (b) `void f(double);`
   (c) `void f(x);`
   (d) `f(double x);`

*9. What will be the output of the following program?

```
#include <stdio.h>

void swap(int a, int b);

int main(void)
{
  int i = 1, j = 2;

  swap(i, j);
  printf("i = %d, j = %d\n", i, j);
  return 0;
}

void swap(int a, int b)
{
  int temp = a;
  a = b;
  b = temp;
}
```

10. Write functions that return the following values. (Assume that `a` and `n` are parameters, where `a` is an array of `int` values and `n` is the length of the array.)

    (a) The largest element in `a`.
    (b) The average of all elements in `a`.
    (c) The number of positive elements in `a`.

11. Write the following function:

    `float compute_GPA(char grades[], int n);`

    The `grades` array will contain letter grades (`A`, `B`, `C`, `D`, or `F`, either upper-case or lower-case); `n` is the length of the array. The function should return the average of the grades (assume that `A` = 4, `B` = 3, `C` = 2, `D` = 1, and `F` = 0).

12. Write the following function:

    `double inner_product(double a[], double b[], int n);`

    The function should return `a[0] * b[0] + a[1] * b[1] + ... + a[n-1] * b[n-1]`.

13. Write the following function, which evaluates a chess position:

    `int evaluate_position(char board[8][8]);`

    `board` represents a configuration of pieces on a chessboard, where the letters `K`, `Q`, `R`, `B`, `N`, `P` represent White pieces, and the letters `k`, `q`, `r`, `b`, `n`, and `p` represent Black pieces. `evaluate_position` should sum the values of the White pieces (`Q` = 9, `R` = 5, `B` = 3, `N` = 3, `P` = 1). It should also sum the values of the Black pieces (done in a similar way). The function will return the difference between the two numbers. This value will be positive if White has an advantage in material and negative if Black has an advantage.

14. The following function is supposed to return `true` if any element of the array `a` has the value 0 and `false` if all elements are nonzero. Sadly, it contains an error. Find the error and show how to fix it:


```
bool has_zero(int a[], int n)
{
  int i;

  for (i = 0; i < n; i++)
    if (a[i] == 0)
      return true;
    else
      return false;
}
```

15. The following (rather confusing) function finds the median of three numbers. Rewrite the function so that it has just one `return` statement.

```
double median(double x, double y, double z)
{
  if (x <= y)
    if (y <= z) return y;
    else if (x <= z) return z;
    else return x;
  if (z <= y) return y;
  if (x <= z) return x;
  return z;
}
```

on 9.6

16. Condense the `fact` function in the same way we condensed `power`.

17. Rewrite the `fact` function so that it's no longer recursive.

18. Write a recursive version of the `gcd` function (see Exercise 3). Here's the strategy to use for computing `gcd(m, n)`: If `n` is 0, return `m`. Otherwise, call `gcd` recursively, passing `n` as the first argument and `m % n` as the second.

*19. Consider the following "mystery" function:

```
void pb(int n)
{
  if (n != 0) {
    pb(n / 2);
    putchar('0' + n % 2);
  }
}
```

Trace the execution of the function by hand. Then write a program that calls the function, passing it a number entered by the user. What does the function do?

Programming Projects

Programming Projects

1. Write a program that asks the user to enter a series of integers (which it stores in an array), then sorts the integers by calling the function `selection_sort`. When given an array with n elements, `selection_sort` must do the following:

   1. Search the array to find the largest element, then move it to the last position in the array.
   2. Call itself recursively to sort the first $n - 1$ elements of the array.

or 12. Any other roll is called the "point" and the game continues. On each subsequent roll, the player wins if he or she rolls the point again. The player loses by rolling 7. Any other roll is ignored and the game continues. At the end of each game, the program will ask the user whether or not to play again. When the user enters a response other than `y` or `Y`, the program will display the number of wins and losses and then terminate.

```
You rolled: 8
Your point is 8
You rolled: 3
You rolled: 10
You rolled: 8
You win!

Play again? y

You rolled: 6
Your point is 6
You rolled: 5
You rolled: 12
You rolled: 3
You rolled: 7
You lose!

Play again? y

You rolled: 11
You win!

Play again? n

Wins: 2  Losses: 1
```

Write your program as three functions: `main`, `roll_dice`, and `play_game`. Here are the prototypes for the latter two functions:

```
int roll_dice(void);
bool play_game(void);
```

`roll_dice` should generate two random numbers, each between 1 and 6, and return their sum. `play_game` should play one craps game (calling `roll_dice` to determine the outcome of each dice roll); it will return `true` if the player wins and `false` if the player loses. `play_game` is also responsible for displaying messages showing the results of the player's dice rolls. `main` will call `play_game` repeatedly, keeping track of the number of wins and losses and displaying the "you win" and "you lose" messages. *Hint:* Use the `rand` function to generate random numbers. See the `deal.c` program in Section 8.2 for an example of how to call `rand` and the related `srand` function.

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 $1, 2, \ldots, n^2$ 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,