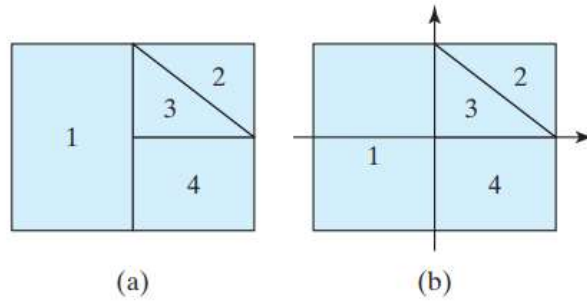


- \*\*5.42** (*Monte Carlo simulation*) A square is divided into four smaller regions as shown in (a). If you throw a dart into the square one million times, what is the probability for the dart to fall into an odd-numbered region? Write a program to simulate the process and display the result. (Hint: Place the center of the square in the center of a coordinate system, as shown in (b). Randomly generate a point in the square and count the number of times for a point to fall in an odd-numbered region.)



**\*\*5.46** (Statistics: compute mean and standard deviation) In business applications, you are often asked to compute the mean and standard deviation of data. The mean is simply the average of the numbers. The standard deviation is a statistic that tells you how tightly all the various data are clustered around the mean in a set of data. For example, what is the average age of the students in a class? How close are the ages? If all the students are the same age, the deviation is 0. Write a program that prompts the user to enter ten numbers, and displays the mean and standard deviations of these numbers using the following formula:

$$\text{mean} = \frac{\sum_{i=1}^n x_i}{n} = \frac{x_1 + x_2 + \dots + x_n}{n}$$
$$\text{deviation} = \sqrt{\frac{\sum_{i=1}^n x_i^2 - \frac{\left(\sum_{i=1}^n x_i\right)^2}{n}}{n - 1}}$$

Here is a sample run:

```
Enter ten numbers: 1 ↵ Enter
2 ↵ Enter
3 ↵ Enter
5.5 ↵ Enter
5.6 ↵ Enter
6 ↵ Enter
7 ↵ Enter
8 ↵ Enter
9 ↵ Enter
10 ↵ Enter
The mean is 5.61
The standard deviation is 2.99794
```

- \*13.4** (*Write/read data*) Write a program that writes 100 integers created randomly into a file. Integers are separated by a space in the file. Read the data back from the file and display the sorted data. Your program should prompt the user to enter a filename. If the file already exists, do not override it. Here is a sample run:

```
Enter a filename: test.txt ↵ Enter
The file already exists
```

```
Enter a filename: test1.txt ↵ Enter
20 34 43 ... 50
```

- \*\*10.22** (*Simulation: coupon collector's problem*) Coupon Collector is a classic statistics problem with many practical applications. The problem is to pick objects from a set of objects repeatedly and find out how many picks are needed for all the objects to be picked at least once. A variation of the problem is to pick cards from a shuffled deck of 52 cards repeatedly and find out how many picks are needed before you see one of each suit. Assume a picked card is placed back in the deck before picking another. Write a program to simulate the number of picks needed to get four cards, one from each suit and display the four cards picked (it is possible a card may be picked twice). Here is a sample run of the program:

```
Queen of Spades
5 of Clubs
Queen of Hearts
4 of Diamonds
Number of picks: 12
```

**\*\*\*10.29** (*Game: hangman*) Write a hangman game that randomly generates a word and prompts the user to guess one letter at a time, as shown in the sample run. Each letter in the word is displayed as an asterisk. When the user makes a correct guess, the actual letter is then displayed. When the user finishes a word, display the number of misses and ask the user whether to continue playing. Create a list to store the words, as follows:

```
# Use any words you wish
words = ["write", "that", "program", ...]
```

```
(Guess) Enter a letter in word ***** > p 
(Guess) Enter a letter in word p***** > r 
(Guess) Enter a letter in word pr**r** > p 
    p is already in the word
(Guess) Enter a letter in word pr**r** > o 
(Guess) Enter a letter in word pro*r** > g 
(Guess) Enter a letter in word progr** > n 
    n is not in the word
(Guess) Enter a letter in word progr** > m 
(Guess) Enter a letter in word progr*m > a 
The word is program. You missed 1 time

Do you want to guess another word? Enter y or n>
```