

## Chapter Two

### Programming Exercise

1. Write a program that displays all the numbers from 100 to 200, ten per line, that are divisible by 5 or 6, but not both. Numbers are separated by exactly one space.
2. Write a program that prompts the user to enter a list and displays whether the list is sorted or not.
3. Write a program that simulates a simple slot machine in which three numbers between 0 and 9 are randomly selected and printed side by side. Print an appropriate statement if all three of the numbers are the same or if any two of the numbers are the same. Continue playing until the user chooses to stop.
4. Suppose that the tuition for a university is \$10,000 this year and increases 5% every year. In one year, the tuition will be \$10,500. Write a program that computes the tuition in ten years and the total cost of four years' worth of tuition after the tenth year.
5. Write a program that randomly fills in 0s and 1s into a 4-by-4 matrix, prints the matrix, and finds the first row and column with the most 1s. Here is a sample run of the program:

```
0011
0011
1101
1010
The largest row index: 2
The largest column index: 2
```

6. Write a program that reads an integer value representing a year from the user. The purpose of the program is to determine whether the year is a leap year (and therefore has 29 days in February) in the Gregorian calendar. A year is a leap year if it is divisible by 4, unless it is also divisible by 100 but not 400. For example, the year 2003 is not a leap year, but 2004 is. The year 1900 is not a leap year because it is divisible by 100, but the year 2000 is a leap year because even though it is divisible by 100, it is also divisible by 400. Produce an error message for any input value less than 1582 (the year the Gregorian calendar was adopted).
7. You have three identical prizes to give away and a pool of 30 finalists. The finalists are assigned numbers from 1 to 30. Write a program to randomly select the numbers of three finalists to receive a prize. Make sure not to pick the same number twice. For example, picking finalists 3, 15, 29 would be valid but picking 3, 3, 31 would be invalid because finalist number 3 is listed twice and 31 is not a valid finalist number.
8. Write a program that plays the Hi-Lo guessing game with numbers. The program should pick a random number between 1 and 100 (inclusive) and then repeatedly prompt the user to guess the number. On each guess, report to the user that he or she is correct or that the guess is high or low. Continue accepting guesses until the user guesses correctly or chooses to quit. Use a sentinel value to determine whether the user wants to quit. Count the number of guesses, and report that value when the user guesses correctly. At the end of each game (by quitting or a correct guess), prompt to determine whether the user wants to play again. Continue playing games until the user chooses to stop.

9. Suppose the weekly hours for all employees are stored in a two-dimensional array. Each row records an employee's seven-day work hours with seven columns. For example, the following array stores the work hours for eight employees. Write a program that displays employees and their total hours in decreasing order of the total hours.

	Su	M	T	W	Th	F	Sa
Employee 0	2	4	3	4	5	8	8
Employee 1	7	3	4	3	3	4	4
Employee 2	3	3	4	3	3	2	2
Employee 3	9	3	4	7	3	4	1
Employee 4	3	5	4	3	6	3	8
Employee 5	3	4	4	6	3	4	4
Employee 6	3	7	4	8	3	8	4
Employee 7	6	3	5	9	2	7	9

10. Write an application that prompts for and reads a double value representing a monetary amount. Then determine the least number of each bill and coin needed to represent that amount, starting with the highest (assume that a ten-dollar bill is the maximum size needed). For example, if the value entered is 47.63 (forty-seven dollars and sixty-three cents), then the program should print the equivalent amount as

```
4 ten dollar bills
1 five dollar bills
2 one dollar bills
2 quarters
1 dimes
0 nickels
3 pennies
```

11. Write a program that creates a histogram that allows you to visually inspect the frequency distribution of a set of values. The program should read in an arbitrary number of integers that are in the range 1 to 100 inclusive; then it should produce a chart similar to the following one that indicates how many input values fell in the range 1 to 10, 11 to 20, and so on.

```
Print one asterisk for each value entered.
1 - 10 | *****
11 - 20 | **
21 - 30 | *****
31 - 40 |
41 - 50 | ***
51 - 60 | *****
61 - 70 | **
81 - 80 | *****
81 - 90 | *****
91 - 100 | *****
```

12. Write a program that repeatedly prompts the user to enter a capital city for African countries. Upon receiving the user input, the program reports whether the answer is correct. The program prompts the user to answer capitals and displays the total correct count. The user's answer is not case-sensitive.
13. The game of Pig is a simple two-player dice game in which the first player to reach 100 or more points wins. Players take turns. On each turn, a player rolls a six-sided die:
  - If the player rolls a 1, then the player gets no new points and it becomes the other player's turn.
  - If the player rolls 2 through 6, then he or she can either
    - ROLL AGAIN or
    - HOLD. At this point, the sum of all rolls is added to the player's score, and it becomes the other player's turn.

Write a program that plays the game of Pig, where one player is a human and the other is the computer. When it is the human's turn, the program should show the score of both players and the previous roll. Allow the human to input "r" to roll again or "h" to hold.

The computer program should play according to the following rule:

- Keep rolling when it is the computer's turn until it has accumulated 20 or more points, then hold. If the computer wins or rolls a 1, then the turn ends immediately.
- Allow the human to roll first.