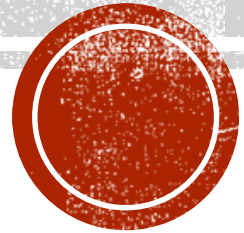
CS 101 LAB #6

ARRAYS



Mythili Vutukuru

IIT Bombay

Reference: "C How to Program", Deitel and Deitel, 8th Edition, Chapter 6

GOAL OF LAB #6

- In this lab, you will write programs to understand the following concepts:
  - Using arrays (one and two dimensional arrays) to solve problems
  - Recursive functions using arrays
- Submit any two programs for this lab, but attempt as many as you can
- Please write code neatly, with proper indentations and comments

USING ARRAYS

6.14 *(Union of Sets)* Use one-dimensional arrays to solve the following problem. Read in two sets of numbers, each having 10 numbers. After reading all values, display all the unique elements in the collection of both sets of numbers. Use the smallest possible array to solve this problem.

6.15 *(Intersection of Sets)* Use one-dimensional arrays to solve the following problem. Read in two sets of numbers, each having 10 numbers. After reading all values, display the unique elements common to both sets of numbers.

6.28 *(Duplicate Elimination)* In Chapter 12, we explore the high-speed binary search tree data structure. One feature of a binary search tree is that duplicate values are discarded when insertions are made into the tree. This is referred to as duplicate elimination. Write a program that produces 20 random numbers between 1 and 20. The program should store all nonduplicate values in an array. Use the smallest possible array to accomplish this task.

USING ARRAYS

6.19 *(Dice Rolling)* Write a program that simulates the rolling of two dice. The program should use rand twice to roll the first die and second die, respectively. The sum of the two values should then be calculated. [*Note:* Because each die can show an integer value from 1 to 6, then the sum of the two values will vary from 2 to 12, with 7 being the most frequent sum and 2 and 12 the least frequent sums.] Figure 6.24 shows the 36 possible combinations of the two dice. Your program should roll the two dice 36,000 times. Use a one-dimensional array to tally the numbers of times each possible sum appears. Print the results in a tabular format. Also, determine if the totals are reasonable; i.e., there are six ways to roll a 7, so approximately one-sixth of all the rolls should be 7.

	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

USING ARRAYS

6.20 *(Game of Craps)* Write a program that runs 1000 games of craps (without human intervention) and answers each of the following questions:

a) How many games are won on the first roll, second roll, …, twentieth roll and after the twentieth roll?

b) How many games are lost on the first roll, second roll, …, twentieth roll and after the twentieth roll?

c) What are the chances of winning at craps? [*Note:* You should discover that craps is one of the fairest casino games. What do you suppose this means?]

d) What's the average length of a game of craps?

e) Do the chances of winning improve with the length of the game?

USING ARRAYS

6.21 An $n \times m$ two-dimensional matrix can be multiplied by another $m \times p$ matrix to give a matrix whose elements are the sum of the products of the elements within a row from the first matrix and the associated elements of a column of the second matrix. Both matrices should either be square matrices, or the number of columns of the first matrix should equal the number of rows of the second matrix.

To calculate each element of the resultant matrix, multiply the first element of a given row from the first matrix and the first element of a given column in the second matrix, add that to the product of the second element of the same row and the second element of the same column, and keep doing so until the last elements of the row and column have been multiplied and added to the sum.

Write a program to calculate the product of 2 matrices and store the result in a third matrix.

USING ARRAYS

6.22 *(Total Sales)* Use a two-dimensional array to solve the following problem. A company has four salespeople (1 to 4) who sell five different products (1 to 5). Once a day, each salesperson passes in a slip for each different type of product sold. Each slip contains:

a) The salesperson number
b) The product number
c) The total dollar value of that product sold that day

Thus, each salesperson passes in between 0 and 5 sales slips per day. Assume that the information from all of the slips for last month is available. Write a program that will read all this information for last month's sales and summarize the total sales by salesperson by product. All totals should be stored in the two-dimensional array `sales`. After processing all the information for last month, print the results in tabular format with each column representing a particular salesperson and each row representing a particular product. Cross total each row to get the total sales of each product for last month; cross total each column to get the total sales by salesperson for last month. Your tabular printout should include these cross totals to the right of the totaled rows and to the bottom of the totaled columns.

USING ARRAYS

6.30 *(The Sieve of Eratosthenes)* A prime integer is any integer greater than 1 that can be divided evenly only by itself and 1. The Sieve of Eratosthenes is a method of finding prime numbers. It works as follows:

a) Create an array with all elements initialized to 1 (true). Array elements with prime indices will remain 1. All other array elements will eventually be set to zero.

b) Starting with array index 2 (index 1 is not prime), every time an array element is found whose value is 1, loop through the remainder of the array and set to zero every element whose index is a multiple of the index for the element with value 1. For array index 2, all elements beyond 2 in the array that are multiples of 2 will be set to zero (indices 4, 6, 8, 10, and so on.). For array index 3, all elements beyond 3 in the array that are multiples of 3 will be set to zero (indices 6, 9, 12, 15, and so on.).

When this process is complete, the array elements that are still set to 1 indicate that the index is a prime number. Write a program that uses an array of 1,000 elements to determine and print the prime numbers between 1 and 999. Ignore element 0 of the array.

ARRAYS AND RECURSION

6.35 *(Print an Array)* Write a recursive function `printArray` that takes an array and the size of the array as arguments, prints the array, and returns nothing. The function should stop processing and return when it receives an array of size zero.

6.37 *(Find the Minimum Value in an Array)* Write a recursive function `recursiveMinimum` that takes an integer array and the array size as arguments and returns the smallest element of the array. The function should stop processing and return when it receives an array of one element.