

Lab Assignment-4

4. Programming project on Repetition and Looping Statements

- 4.1 An integer n is divisible by 9 if the sum of its digits is divisible by 9. Develop a program to display each digit, starting with the rightmost digit. Your program should also determine whether or not the number is divisible by 9. Test it on the following numbers:

```
n= 154368
n= 621594
n= 123456
```

Hint: Use the % operator to get each digit; then use / to remove that digit. So 154368 % 10 gives 8 and 154368 / 10 gives 15436. The next digit extracted should be 6, then 3 and so on.

- 4.2 Write a program to create an output file containing a customized loan amortization table. Your program will prompt the user to enter the amount borrowed (the principal), the annual interest rate, and the number of payments (n). To calculate the monthly payment, it will use the formula from question 3.1 in Assignment 3. This payment must be rounded to the nearest cent. After the payment has been rounded to the nearest cent, the program will write to the output file n lines showing how the debt is paid off. Each month part of the payment is the monthly interest on the principal balance, and the rest is applied to the principal. Because the payment and each month's interest are rounded, the final payment will be a bit different and must be calculated as the sum of the final interest payment and the final principal balance. Here is a sample table for a \$1000 loan borrowed at a 9% annual interest rate and paid back over 6 months.

Principal	\$1000.00	Payment	\$171.07
Annual interest	9.0%	Term	6 months
<hr/>			
Payment	Interest	Principal	Principal Balance
1	7.50	163.57	836.43
2	6.27	164.80	671.63
3	5.04	166.03	505.60
4	3.79	167.28	338.32
5	2.54	168.53	169.79
6	1.27	169.79	0.00
<hr/>			
Final payment	\$171.06		

- 4.3
- Write a program that will find the smallest, largest, and average values in a collection of N numbers. Get the value of N before scanning each value in the collection of N numbers.
 - Modify your program to compute and display both the range of values in the data collection and the standard deviation of the data collection. To compute the standard deviation, accumulate the sum of the squares of the data values (`sum_squares`) in the main loop. After loop exit, use the formula

$$\text{standard deviation} = \sqrt{\frac{\text{sum_squares}}{N} - \text{average}^2}$$

- 4.4 The greatest common divisor (gcd) of two integers is the product of the integers' common factors. Write a program that inputs two numbers and implements the following approach to finding their gcd. We will use the numbers -252 and 735. Working with the numbers' absolute values, we find the remainder of one divided by the other.

$$\begin{array}{r} 0 \\ 735 \overline{)252} \end{array}$$

Now we calculate the remainder of the old divisor divided by the remainder found.

$$\begin{array}{r} 2 \\ 252 \overline{)735} \\ \underline{504} \\ 231 \end{array}$$

We repeat this process until the remainder is zero.

$$\begin{array}{r} 1 \\ 231 \overline{)252} \\ \underline{231} \\ 21 \end{array} \qquad \begin{array}{r} 11 \\ 21 \overline{)231} \\ \underline{21} \\ 21 \\ \underline{21} \\ 0 \end{array}$$

The last divisor (21) is the gcd.

- 4.5 The Environmental Awareness Club of BigCorp International is proposing that the company subsidize at \$.08 per passenger-kilometer the commuting costs of employees who form carpools that meet a prescribed minimum passenger efficiency. Passenger efficiency P (in passenger-kilometers per liter) is defined as

$$P = \frac{ns}{l}$$

where n is the number of passengers, s is the distance traveled in kilometers, and l is the number of liters of gasoline used.

Write a program that processes an input file of data on existing carpools (carpool.txt), creating an output file (effic.txt) containing a table of all carpools that meet the passenger efficiency minimum. The input file begins with a number that is the minimum passenger efficiency. Each carpool is represented by a data line containing three numbers: the number of people in the carpool, the total commuting distance per five-day week, and the number of liters of gasoline consumed in a week of commuting. The data file ends with a line of zeros. Write your results with this format:

CARPOOLS MEETING MINIMUM PASSENGER EFFICIENCY OF 25 PASSENGER KM/L

Passengers	Weekly Commute (km)	Gasoline Consumption(L)	Efficiency (pass km/L)	Weekly Subsidy(\$)
4	75	11.0	27.3	24.00
2	60	4.5	26.7	9.60
...				

- 4.6 a. Write a program to process a collection of daily high temperatures. Your program should count and print the number of hot days (high temperature 85 or higher), the number of pleasant days (high temperature 60–84), and the number of cold days (high temperatures less than 60). It should also display the category of each temperature. Test your program on the following data:

55 62 68 74 59 45 41 58 60 67 65 78 82 88 91 92 90
 93 87 80 78 79 72 68 61 59

b. Modify your program to display the average temperature (a real number) at the end of the run.

4.7 Write a program to process weekly employee time cards for all employees of an organization. Each employee will have three data items: an identification number, the hourly wage rate, and the number of hours worked during a given week. Each employee is to be paid time and a half for all hours worked over 40. A tax amount of 3.625% of gross salary will be deducted. The program output should show the employee's number and net pay. Display the total payroll and the average amount paid at the end of the run.

4.8 Suppose you own a beer distributorship that sells Piels (ID number 1), Coors (ID number 2), Bud (ID number 3), and Iron City (ID number 4) by the case. Write a program to

- Get the case inventory for each brand for the start of the week.
- Process all weekly sales and purchase records for each brand.
- Display out the final inventory.

Each transaction will consist of two data items. The first item will be the brand ID number (an integer). The second will be the amount purchased (a positive integer value) or the amount sold (a negative integer value). For now you may assume that you always have sufficient foresight to prevent depletion of your inventory for any brand. (Hint: Your data entry should begin with four values representing the case inventory, followed by the transaction values.)

4.9 The pressure of a gas changes as the volume and temperature of the gas vary. Write a program that uses the Van der Waals equation of state for a gas,

$$\left(P + \frac{an^2}{V^2}\right)(V - bn) = nRT$$

to create a file that displays in tabular form the relationship between the pressure and the volume of n moles of carbon dioxide at a constant absolute temperature (T). P is the pressure in atmospheres, and V is the volume in liters. The Van der Waals constants for carbon dioxide are $a = 3.592 \text{ L}^2 \cdot \text{atm/mol}^2$ and $b = 0.0427 \text{ L/mol}$. Use $0.08206 \text{ L} \cdot \text{atm/mol} \cdot \text{K}$ for the gas constant R . Inputs to the program include n , the Kelvin temperature, the initial and final volumes in milliliters, and the volume increment between lines of the table. Your program will output a table that varies the volume of the gas from the initial to the final volume in steps prescribed by the volume increment. Here is a sample run:

Please enter at the prompts the number of moles of carbon dioxide, the absolute temperature, the initial volume in milliliters, the final volume, and the increment volume between lines of the table.

Quantity of carbon dioxide (moles)> 0.02

Temperature (kelvin)> 300

Initial volume (milliliters)> 400

Final volume (milliliters)> 600

Volume increment (milliliters)> 50

Output File

0.0200 moles of carbon dioxide at 300 kelvin

Volume (ml)	Pressure (atm)
-------------	----------------

400	1.2246
450	1.0891
500	0.9807
550	0.8918
600	0.8178

- 4.10 A concrete channel to bring water to Crystal Lake is being designed. It will have vertical walls and be 15 feet wide. It will be 10 feet deep, have a slope of .0015 feet/foot, and a roughness coefficient of .014. How deep will the water be when 1,000 cubic feet per second is flowing through the channel? To solve this problem, we can use Manning's equation:

$$Q = \frac{1.486}{N} A R^{2/3} S^{1/2}$$

where Q is the flow of water (cubic feet per second), N is the roughness coefficient (unitless), A is the area (square feet), S is the slope (feet/foot), and R is the hydraulic radius (feet). The hydraulic radius is the cross-sectional area divided by the wetted perimeter. For square channels like the one in this example,

$$\text{Hydraulic radius} = \text{depth} \times \text{width} / (2.0 \times \text{depth} + \text{width})$$

To solve this problem, design a program that allows the user to guess a depth and then calculates the corresponding flow. If the flow is too little, the user should guess a depth a little higher; if the flow is too high, the user should guess a depth a little lower. The guessing is repeated until the computed flow is within 0.1% of the flow desired. To help the user make an initial guess, the program should display the flow for half the channel depth. Note the example run:

At a depth of 5.0000 feet, the flow is 641.3255 cubic feet per second.

Enter your initial guess for the channel depth when the flow is 1000.0000 cubic feet per second
Enter guess> 6.0

Depth: 6.0000 Flow: 825.5906 cfs Target: 1000.0000 cfs Difference: 174.4094 Error: 17.4409 percent
Enter guess> 7.0

Depth: 7.0000 Flow: 1017.7784 cfs Target: 1000.0000 cfs Difference: -17.7784 Error: -1.7778 percent
Enter guess> 6.8

- 4.11 Assume that United States consumers put \$51 billion in fast food charges on their credit and debit cards in 2006, up from \$33.2 billion in 2005. Based on this model of the billions of fast food charges,

$$F(t) = 33.2 + 16.8t$$

where t is years since 2005, write a program that repeatedly prompts the user to enter a year after 2005 and then predicts the billions of dollars of fast food charges U.S. consumers will make in that year. Define and call a function *fast_food_billions* that takes the year as its

input argument and returns the prediction as its result. Tell the user that entry of a year before 2005 will cause the program to stop.

- 4.12 A baseball player's batting average is calculated as the number of hits divided by the official number of at-bats. In calculating official at-bats, walks, sacrifices, and occasions when hit by the pitch are not counted. Write a program that takes an input file containing player numbers and batting records. Trips to the plate are coded in the batting record as follows: H—hit, O—out, W— walk, S—sacrifice, P—hit by pitch. The program should output for each player the input data followed by the batting average. (*Hint* : Each batting record is followed by a newline character.)

Sample input file:

```
12 HOOOWSHHOOHPWWHO
4 OSOHHHWWOHOHOOO
7 WPOHOOHWOHHOWOO
```

Corresponding output:

Player 12's record: HOOOWSHHOOHPWWHO

Player 12's batting average: 0.455

Player 4's record: OSOHHHWWOHOHOOO

Player 4's batting average: 0.417

Player 7's record: WPOHOOHWOHHOWOO

Player 7's batting average: 0.364

- 4.13 The rate of decay of a radioactive isotope is given in terms of its half-life H , the time lapse required for the isotope to decay to one-half of its original mass. The isotope cobalt-60 (^{60}Co) has a half-life of 5.272 years. Compute and print in table form the amount of this isotope that remains after each year for 5 years, given the initial presence of an amount in grams. The value of amount should be provided interactively. The amount of ^{60}Co remaining can be computed by using the following formula:

$$r = \text{amount} \times C^{y/H}$$

where amount is the initial amount in grams, C is expressed as $e^{-0.693}$ ($e = 2.71828$), y is the number of years elapsed, and H is the half-life of the isotope in years.

- 4.14 The value for π can be determined by the series equation:

$$\pi = 4 \times \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - \dots \right)$$

Write a program to approximate the value of π using the formula given including terms up through $1/99$.

- 4.15 In this chapter we studied the bisection method for finding a root of an equation. Another method for finding a root, Newton's method, usually converges to a solution even faster than the bisection method, if it converges at all. Newton's method starts with an initial guess for a root, x_0 , and then generates successive approximate roots $x_1, x_2, \dots, x_j, x_{j+1}, \dots$, using the iterative formula

$$x_{j+1} = x_j - \frac{f(x_j)}{f'(x_j)}$$

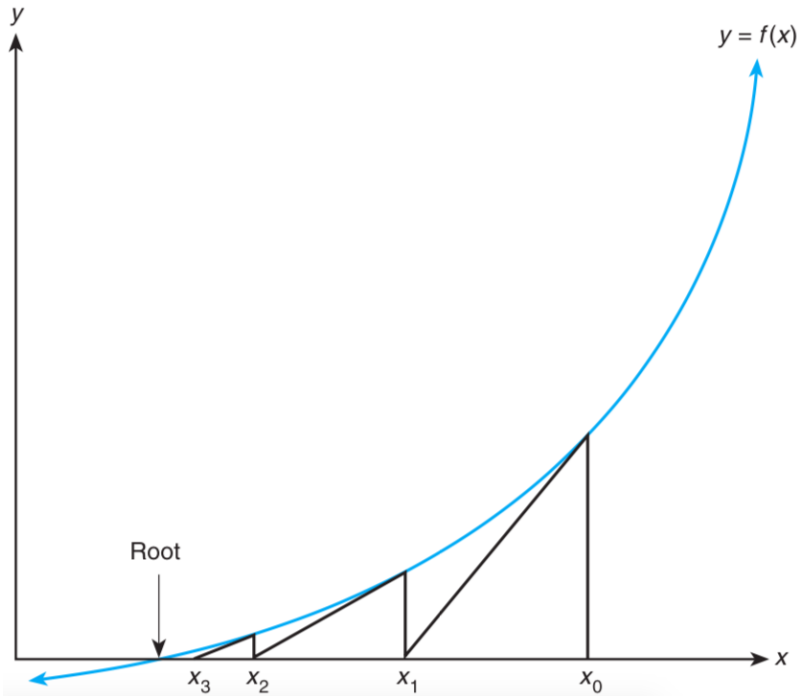


Figure 1: Geometric Interpretation of Newton's Method

where $f'(x)$ is the derivative of function f evaluated at $x = x_j$. The formula generates a new guess, x_{j+1} , from a previous one, x_j . Sometimes Newton's method will fail to converge to a root. In this case, the program should terminate after many trials, perhaps 100.

Figure 1 shows the geometric interpretation of Newton's method where x_0, x_1 , and x_2 represent successive guesses for the root. At each point x_j , the derivative, $f'(x_j)$, is the slope of the tangent to the curve, $f(x)$. The next guess for the root, x_{j+1} , is the point at which the tangent crosses the x-axis. From geometry, we get the equation

$$\frac{y_{j+1} - y_j}{x_{j+1} - x_j} = m$$

where m is the slope of the line between points (x_{j+1}, y_{j+1}) and (x_j, y_j) . In Fig. 1, we see that y_{j+1} is zero, y_j is $f(x_j)$, and m is $f'(x_j)$ therefore, by substituting and rearranging terms, we get

$$-f(x_j) = f'(x_j) \times (x_{j+1} - x_j)$$

leading to the formula shown at the beginning of this problem.

Write a program that uses Newton's method to approximate the n th root of a number to six decimal places. If $x^n = c$, then $x^n - c = 0$. Finding a root of the second equation will give you $\sqrt[n]{c}$. Test your program on $\sqrt{2}$, $\sqrt[3]{7}$, and $\sqrt[3]{-1}$. Your program could use $c/2$ as its initial guess.

4.16 You would like to find the area under the curve

$$y = f(x)$$

between the lines $x = a$ and $x = b$. One way to approximate this area is to use line segments as approximations of small pieces of the curve and then to sum the areas of trapezoids created by drawing perpendiculars from the line segment endpoints to the x -axis, as shown in Fig. 2. We will assume that $f(x)$ is non-negative over the interval $[a, b]$. The trapezoidal rule approximates

this area T as

$$T = \frac{h}{2} \left(f(a) + f(b) + 2 \sum_{i=1}^{n-1} f(x_i) \right)$$

for n subintervals of length h :

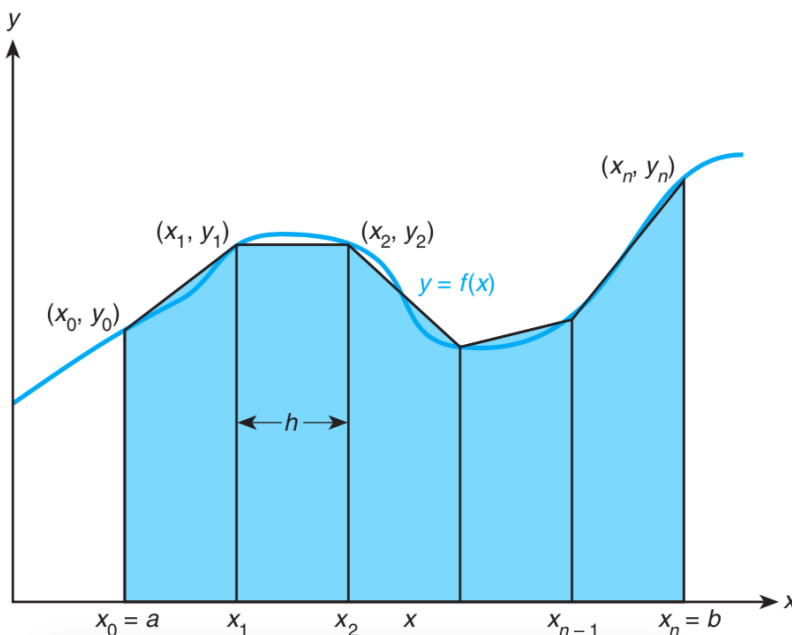


Figure 2: Approximating the Area Under a Curve with Trapezoids

$$h = \frac{b - a}{n}$$

Write a function *trap* with input parameters a, b, n , and f that implements the trapezoidal rule. Call *trap* with values for n of 2, 4, 8, 16, 32, 64, and 128 on functions

$$g(x) = x^2 \sin x \quad (a = 0, b = 3.14159)$$

and

$$h(x) = \sqrt{4 - x^2} \quad (a = -2, b = 2)$$

Function h defines a half-circle of radius 2. Compare your approximation to the actual area of this half-circle.

Note : If you have studied calculus, you will observe that the trapezoidal rule is approximating

$$\int_a^b f(x) dx$$