LAB ASSIGNMENTS

Problem Solving and Program Design Using C

(CSE 3942)

Department of Computer Science & Information Technology
Faculty of Engineering & Technology (ITER)
Siksha 'O' Anusandhan Deemed To Be University
Bhubaneswar, Odisha - 751030

Lab Assignment-1

1. Programming Projects on Overview of C

1.1 Write a program that calculates mileage reimbursement for a salesperson at a rate of $.35 per mile. Your program should interact with the user in this manner:

MILEAGE REIMBURSEMENT CALCULATOR
Enter beginning odometer reading=> 13505.2
Enter ending odometer reading=> 13810.6
You traveled 305.4 miles. At $0.35 per mile,
your reimbursement is $106.89.

1.2 Write a program to assist in the design of a hydroelectric dam. Prompt the user for the height of the dam and for the number of cubic meters of water that are projected to flow from the top to the bottom of the dam each second. Predict how many megawatts (1MW = 10^6W) of power will be produced if 90% of the work done on the water by gravity is converted to electrical energy. Note that the mass of one cubic meter of water is 1000 kg. Use 9.80 $meters/second^2$ as the gravitational constant g . Be sure to use meaningful names for both the gravitational constant and the 90% efficiency constant. For one run, use a height of 170 m and flow of 1.30×10^3 m^3/s. The relevant formula (w = work, m = mass, g = gravity, h = height) is: w = mgh.

1.3 Write a program that estimates the temperature in a freezer (in °C) given the elapsed time (hours) since a power failure. Assume this temperature (T) is given by

$$T = \frac{4t^2}{t+2} - 20$$

where t is the time since the power failure. Your program should prompt the user to enter how long it has been since the start of the power failure in whole hours and minutes. Note that you will need to convert the elapsed time into hours. For example, if the user entered 2 30 (2 hours 30 minutes), you would need to convert this to 2.5 hours.

1.4 Write a program to convert a temperature in degrees Fahrenheit to degrees Celsius.

DATA REQUIREMENTS
Problem Input
int fahrenheit /* temperature in degrees Fahrenheit */
Problem Output
double celsius /* temperature in degrees Celsius */
Relevant Formula
celsius = 5/9 (fahrenheit - 32)

1.5 Hospitals use programmable pumps to deliver medications and fluids to intravenous lines at a set number of milliliters per hour. Write a program to output information for the labels the hospital pharmacy places on bags of I.V. medications indicating the volume of medication to be infused and the rate at which the pump should be set. The program should prompt the user to enter the quantity of fluid in the bag and the number of minutes over which it should be infused. Output the VTBI (volume to be infused) in ml and the infusion rate in ml/hr.

Sample run:

Volume to be infused (ml) => 100
Minutes over which to infuse => 20

VTBI: 100 ml
Rate: 300 ml/hr

1.6 Write a program that predicts the score needed on a final exam to achieve a desired grade in a course. The program should interact with the user as follows:

Enter desired grade> B
Enter minimum average required> 79.5
Enter current average in course> 74.6
Enter how much the final counts
as a percentage of the course grade> 25

You need a score of 94.20 on the final to get a B.

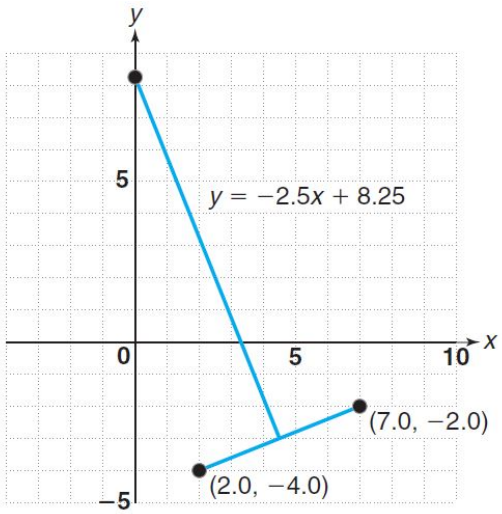
In the example shown, the final counts 25 percent of the course grade.

1.7 Write a program that calculates how many BTUs of heat are delivered to a house given the number of gallons of oil burned and the efficiency of the house's oil furnace. Assume that a barrel of oil (42 gallons) has an energy equivalent of 5,800,000 BTU. (Note: This number is too large to represent as an int on some personal computers.) For one test use an efficiency of 65 percent and 100 gallons of oil.

1.8 Metro City Planners proposes that a community conserve its water supply by replacing all the community's toilets with low-flush models that use only 2 liters per flush. Assume that there is about 1 toilet for every 3 persons, that existing toilets use an average of 15 liters per flush, that a toilet is flushed on average 14 times per day, and that the cost to install each new toilet is $150. Write a program that would estimate the magnitude (liters/day) and cost of the water saved based on the community's population.

1.9 Write a program that takes the length and width of a rectangular yard and the length and width of a rectangular house situated in the yard. Your program should compute the time required to cut the grass at the rate of two square feet a second.

1.10 Write a program that outputs the equation of the perpendicular bisector of the line segment between two points. Your program should



- prompt for and input the coordinates of the two points [for example, try the points (2.0, -4.0) and (7.0, -2.0)];
- compute the slope of the line between those two points;
- compute the coordinates of the midpoint of the line segment between the two points by averaging the two x coordinates and the two y coordinates;
- compute the slope of the perpendicular bisector by taking the negative reciprocal of the slope of the line segment;
- compute the y intercept of the perpendicular bisector (you now have the slope m of the bisector and a point (x_{mid} , y_{mid}) on the bisector, so the y intercept is y_{mid} - m x_{mid}); and
- output with labels the original two points, and output in y = mx + b format the equation of the perpendicular bisector. The below mentioned Figure illustrates the sample line segment mentioned above and its perpendicular bisector.

Test your program to be sure it works on different pairs of points. However, there will be some pairs of points for which you can't make your program work (at least not at this stage). Think about what points will cause your program to fail, and write a paragraph describing which points fall in this category.

1.11 The Pythagorean theorem states that the sum of the squares of the sides of a right triangle is equal to the square of the hypotenuse. For example, if two sides of a right triangle have lengths of 3 and 4, then the hypotenuse must have a length of 5. Together the integers 3, 4, and 5 form a Pythagorean triple. There are an infinite number of such triples. Given two positive integers, m and n, where m > n, a Pythagorean triple can be generated by the following formulas:

$$side1 = m^2 - n^2$$
$$side2 = 2mn$$
$$hypotenuse = m^2 + n^2$$

The triple (side1 = 3, side2 = 4, hypotenuse = 5) is generated by this formula when m = 2 and n = 1. Write a program that takes values for m and n as input and displays the values of the Pythagorean triple generated by the formulas above.

1.12 Write a program that calculates the acceleration (m/s^2) of a jet fighter launched from an aircraft-carrier catapult, given the jet's takeoff speed in km/hr and the distance (meters) over which the catapult accelerates the jet from rest to takeoff. Assume constant acceleration. Also calculate the time (seconds) for the fighter to be accelerated to takeoff speed. When you prompt the user, be sure to indicate the units for each input. For one run, use a takeoff speed of 278 km/hr and a distance of 94 meters. Relevant formulas (v = velocity, a = acceleration, t = time, s = distance)

$$v = at$$
$$s = \frac{1}{2}at^2$$

Lab Assignment-2

2. Programming project on Top-Down Design With Functions

2.1 You have saved $500 to use as a down payment on a car. Before beginning your car shopping, you decide to write a program to help you figure out what your monthly payment will be, given the car's purchase price, the monthly interest rate, and the time period over which you will pay back the loan. The formula for calculating your payment is

$$\text{payment} = \frac{iP}{1-(1+i)^{-n}}$$

where P = principal (the amount you borrow)
i = monthly interest rate $\left(\frac{1}{12}\right.$ of the annual rate)
n = total number of payments

Your program should prompt the user for the purchase price, the down payment, the annual interest rate and the total number of payments (usually 36, 48, or 60). It should then display the amount borrowed and the monthly payment including a dollar sign and two decimal places.

2.2 Write two functions, one that displays a triangle and one that displays a rectangle. Use these functions to write a complete C program from the following outline:

```
int
main(void)
{
/* Draw triangle. */
/* Draw rectangle. */
/* Display 2 blank lines. */
/* Draw triangle. */
/* Draw rectangle. */
}
```

2.3 Add the functions from Fig. 3.14 to the ones for Programming Project 2 from the Text Book. Use these functions in a program that draws a rocket ship (triangle over rectangles over intersecting lines), a male stick figure (circle over rectangle over intersecting lines), and a female stick figure (circle over triangle over intersecting lines) standing on the head of a male stick figure. Write function skip_5_lines and call it to place five blank lines between drawings.

2.4 For any integer $n > 0, n\,!$ is defined as the product $n \times n-1 \times n-2 \ldots \times 2 \times 1.0\,!$ is defined to be 1 . It is sometimes useful to have a closed-form definition instead; for this purpose, an approximation can be used. R.W. Gosper proposed the following such approximation formula:

$$n! \approx n^n e^{-n} \sqrt{\left(2n + \frac{1}{3}\right)\pi}$$

Create a program that prompts the user to enter an integer n, uses Gosper's formula to approximate $n\,!$, and then displays the result. The message displaying the result should look something like this:

5! equals approximately 119.97003

Your program will be easier to debug if you use some intermediate values instead of trying

to compute the result in a single expression. If you are not getting the correct results, then you can compare the results of your intermediate values to what you get when you do the calculations by hand. Use at least two intermediate variables - one for $2n + \frac{1}{3}$ and one for $\sqrt{\left(2n + \frac{1}{3}\right)\pi}$. Display each of these intermediate values to simplify debugging. Be sure to use a named constant for PI, and use the approximation 3.14159265. Test the program on nonnegative integers less than 8.

2.5 Write a program that takes a positive number with a fractional part and rounds it to two decimal places. For example, 32.4851 would round to 32.49, and 32.4431 would round to 32.44. (Hint: See "Rounding a number" in Table 2.12 and function scale in Fig. 3.23 from the Text Book.)

2.6 Four track stars have entered the mile race at the Penn Relays. Write a program that scans in the race time in minutes (minutes) and seconds (seconds) for a runner and computes and displays the speed in feet per second (fps) and in meters per second (mps). (Hints: There are 5,280 feet in one mile, and one kilometer equals 3,282 feet.) Write and call a function that displays instructions to the program user. Run the program for each star's data.

Minutes	Seconds
3	52.83
3	59.83
4	00.03
4	16.22

2.7 In shopping for a new house, you must consider several factors. In this problem the initial cost of the house, the estimated annual fuel costs, and the annual tax rate are available. Write a program that will determine the total cost of a house after a five-year period and run the program for each of the following sets of data.

Initial House Cost	Annual Fuel Cost	Tax Rate
67,000	2,300	0.025
62,000	2,500	0.025
75,000	1,850	0.020

To calculate the house cost, add the initial cost to the fuel cost for five years, then add the taxes for five years. Taxes for one year are computed by multiplying the tax rate by the initial cost. Write and call a function that displays instructions to the program user.

2.8 A cyclist coasting on a level road slows from a speed of 10 mi/hr to 2.5 mi/hr in one minute. Write a computer program that calculates the cyclist's constant rate of acceleration and determines how long the cyclist will take to come to rest, given an initial speed of 10 mi/hr. (Hint: Use the equation

$$a = \frac{v_f - v_t}{t}$$

where a is acceleration, t is time interval, v_1 is initial velocity, and v_f is final velocity.) Write and call a function that displays instructions to the program user and a function that computes a, given t, v_f, and v_r.

2.9 A manufacturer wishes to determine the cost of producing an open-top cylindrical container. The surface area of the container is the sum of the area of the circular base plus the area of the outside (the circumference of the base times the height of the container). Write a program to take the radius of the base, the height of the container, the cost per square centimeter of the material (cost), and the number of containers to be produced (quantity). Calculate the cost of each container and the total cost of producing all the containers. Write and call a function that displays instructions to the user and a function that computes surface area.

2.10 Write a program to take a depth (in kilometers) inside the earth as input data; compute and display the temperature at this depth in degrees Celsius and degrees Fahrenheit. The relevant formulas are

Celsius = 10 (depth) + 20 (Celsius temperature at depth in km)
Fahrenheit = 1.8 (Celsius) + 32

Include two functions in your program. Function *celsius_at_depth* should compute and return the Celsius temperature at a depth measured in kilometers. Function *fahrenheit* should convert a Celsius temperature to Fahrenheit.

2.11 The ratio between successive speeds of a six-speed gearbox (assuming that the gears are evenly spaced to allow for whole teeth) is

$$\sqrt[5]{M/m}$$

where M is the maximum speed in revolutions per minute and m is the minimum speed. Write a function *speeds_ratio* that calculates this ratio for any maximum and minimum speeds. Write a main function that prompts for maximum and minimum speeds (rpm), calls *speeds_ratio* to calculate the ratio, and displays the results in a sentence of the form

The ratio between successive speeds of a six-speed gearbox with maximum speed ____ rpm and minimum speed ____ rpm is ____ .

2.12 Write a program that calculates the speed of sound (a) in air of a given temperature T (°F). Formula to compute the speed in ft/sec :

$$a = 1086\sqrt{\frac{5T + 297}{247}}$$

Be sure your program does not lose the fractional part of the quotient in the formula shown. As part of your solution, write and call a function that displays instructions to the program user.

2.13 After studying the population growth of Gotham City in the last decade of the 20th century, we have modeled Gotham's population function as

$$P(t) = 52.966 + 2.184t$$

where t is years after 1990, and P is population in thousands. Thus, $P(0)$ represents the population in 1990, which was 52.966 thousand people. Write a program that defines a function named *population* that predicts Gotham's population in the year provided as an input argument. Write a program that calls the function and interacts with the user as follows:

Enter a year after 1990 2015
Predicted Gotham City population for 2010 (in thousands):
107.566

Lab Assignment-3

3. Programming project on Selection Structures: if and switch statements

---

3.1 Keith's Sheet Music needs a program to implement its music teacher's discount policy. The program is to prompt the user to enter the purchase total and to indicate whether the purchaser is a teacher. The store plans to give each customer a printed receipt, so your program is to create a nicely formatted file called receipt.txt. Music teachers receive a 10% discount on their sheet music purchases unless the purchase total is $100 or higher. In that case, the discount is 12%. The discount calculation occurs before addition of the 5% sales tax. Here are two sample output files—one for a teacher and one for a nonteacher.

Total purchases	$122.00
Teacher's discount(12%)	14.64
Discounted total	107.36
Sales tax (5%)	5.37
Total	$112.73

Total purchases	$24.90
Sales tax (5%)	1.25
Total	$26.15

Note: to display a % sign, place two % signs in the format string:
printf("%d%%", SALES_TAX);

3.2 Write a program that calculates the user's body mass index (BMI) and categorizes it as underweight, normal, overweight, or obese, based on the following table from the United States Centers for Disease Control:

BMI	Weight Status
Below 18.5	Underweight
18.5 – 24.9	Normal
25.0 – 29.9	Overweight
30.0 and above	Obese

To calculate BMI based on weight in pounds (wt_lb) and height in inches (ht_in), use this formula (rounded to tenths):

$$\frac{703 \times wt\_lb}{ht\_in^2}$$

Prompt the user to enter weight in pounds and height in inches.

3.3 While spending the summer as a surveyor's assistant, you decide to write a program that transforms compass headings in degrees (0 to 360) to compass bearings. A compass bearing consists of three items: the direction you face (north or south), an angle between 0 and 90 degrees, and the direction you turn before walking (east or west). For example, to get the bearing for a compass heading of 110.0 degrees, you would first face due south (180 degrees) and then turn 70.0 degrees east (180.0 - 70.0 = 110.0). Therefore, the bearing is South 70.0 degrees East. Be sure to check the input for invalid compass headings.

3.4 Write a program that reports the contents of a compressed-gas cylinder based on the first letter of the cylinder's color. The program input is a character representing the observed color of the

cylinder: 'Y' or 'y' for yellow, 'O' or 'o' for orange, and so on. Cylinder colors and associated contents are as follows:

orange	ammonia
brown	carbon monoxide
yellow	hydrogen
green	oxygen

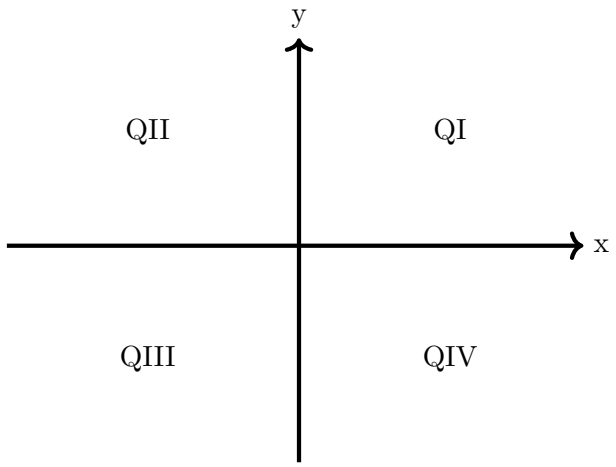
Your program should respond to input of a letter other than the first letters of the given colors with the message, Contents unknown .

3.5 The National Earthquake Information Center has asked you to write a program implementing the following decision table to characterize an earthquake based on its Richter scale number.

Richter Scale Number (n)	Characterization
$n < 5.0$	Little or no damage
$5.0 \leq n < 5.5$	Some damage
$5.5 \leq n < 6.5$	Serious damage: walls may crack or fall
$6.5 \leq n < 7.5$	Disaster: houses and buildings may collapse
higher	Catastrophe: most buildings destroyed

Could you handle this problem with a switch statement? If so, use a switch statement; if not, explain why.

3.6 Write a program that takes the $x - y$ coordinates of a point in the Cartesian plane and prints a message telling either an axis on which the point lies or the quadrant in which it is found.



Sample lines of output:
(-1.0, -2.5) is in quadrant III
(0.0, 4.8) is on the y-axis.

3.7 Write a program that determines the day number (1 to 366) in a year for a date that is provided as input data. As an example, January 1, 1994, is day 1. December 31, 1993, is day 365. December 31, 1996, is day 366, since 1996 is a leap year. A year is a leap year if it is divisible by four, except that any year divisible by 100 is a leap year only if it is divisible by 400. Your program should accept the month, day, and year as integers. Include a function leap that returns 1 if called with a leap year, 0 otherwise.

3.8 Write a program that interacts with the user like this:
(1) Carbon monoxide

(2) Hydrocarbons
(3) Nitrogen oxides
(4) Nonmethane hydrocarbons

Enter pollutant number: 2
Enter number of grams emitted per mile: 0.35
Enter odometer reading: 40112
Emissions exceed permitted level of 0.31 grams/mile.

Use the table of emissions limits below to determine the appropriate message.

	First 50,000 Miles	Second 50,000 Miles
carbon monoxide	3.4 grams/mile	4.2 grams/mile
hydrocarbons	0.31 grams/mile	0.39 grams/mile
nitrogen oxides	0.4 grams/mile	0.5 grams/mile
nonmethane hydrocarbons	0.25 grams/mile	0.31 grams/mile

3.9 Chatflow Wireless offers customers 600 weekday minutes for a flat rate of 39.99. Night (8 P.M. to 7 A.M.) and weekend minutes are free, but additional weekday minutes cost 0.40 each. There are taxes of 5.25Write a program that prompts the user to enter the number of weekday minutes, night minutes, and weekend minutes used, and calculates the monthly bill and average cost of a minute before taxes. The program should display with labels all the input data, the pretax bill and average minute cost, the taxes, and the total bill. Store all monetary values as whole cents (rounding the taxes and average minute cost), and divide by 100 for display of results.

3.10 Write a program to control a bread machine. Allow the user to input the type of bread as W for White and S for Sweet. Ask the user if the loaf size is double and if the baking is manual. The following table details the time chart for the machine for each bread type. Display a statement for each step. If the loaf size is double, increase the baking time by 50 percent. If baking is manual, stop after the loaf-shaping cycle and instruct the user to remove the dough for manual baking. Use functions to display instructions to the user and to compute the baking time.

Operation	White Bread	Sweet Bread
Primary kneading	15 mins	20 mins
Primary rising	60 mins	60 mins
Secondary kneading	18 mins	33 mins
Secondary rising	20 mins	30 mins
Loaf shaping	2 seconds	2 seconds
Final rising	75 mins	75 mins
Baking	45 mins	35 mins
Cooling	30 mins	30 mins

3.11 The table below shows the normal boiling points of several substances. Write a program that prompts the user for the observed boiling point of a substance in °C and identifies the substance if the observed boiling point is within 5% of the expected boiling point. If the data input is more than 5% higher or lower than any of the boiling points in the table, the program should output the message Substance Unknown.

Substance	Normal boiling point (°C)
Water	100
Mercury	357
Copper	1187
Silver	2193
Gold	2660

Your program should define and call a function within_x_percent that takes as parameters a reference value ref , a data value data , and a percentage value x and returns 1 meaning true if data is within x % of ref —that is, (ref – x% * ref) ≤ data ≤ (ref + x % * ref) . Otherwise within_x_percent would return zero, meaning false. For example, the call within_x_percent(357, 323, 10) would return true, since 10% of 357 is 35.7, and 323 falls between 321.3 and 392.7.

3.12 Let's say you are an accountant setting up a payroll system based on Table given below, which shows five different ranges for salaries up to $150,000.00. Each table line shows the base tax amount (column 2) and tax percentage (column 3) for a particular salary range (column 1). For the given person's salary, write a C program which calculates the tax due by adding the base tax to the product of the percentage times the excess salary over the minimum salary for that range.

Salary Range ($)	Base Tax ($)	Percentage of Excess
0.00–14,999.99	0.00	15
15,000.00–29,999.99	2,250.00	18
30,000.00–49,999.99	5,400.00	22
50,000.00–79,999.99	11,000.00	27
80,000.00–150,000.00	21,600.00	33

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

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
Final payment	$171.06		

4.3 a. 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.

b. 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

$$standard\ deviation = \sqrt{\frac{sum\_squares}{N} - average^2}$$

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

a. Get the case inventory for each brand for the start of the week.

b. Process all weekly sales and purchase records for each brand.

c. 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 pres- sure 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,

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

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:

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

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:

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

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 $x_1, x_2, ..., x_j, x_{j+1}, ...$, using the iterative formula

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

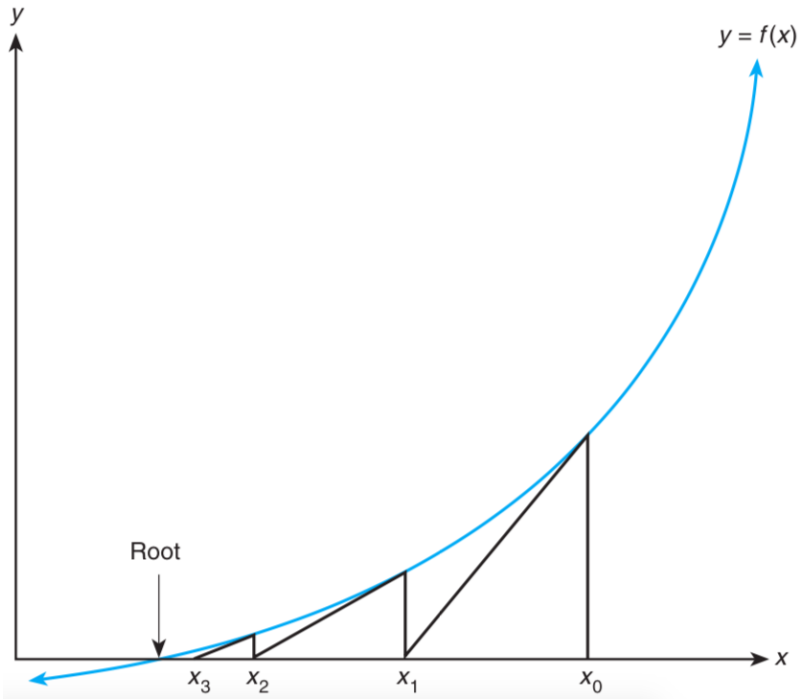


Figure 1: Geometric Interpretation of Newton's Method

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 nth 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. 3. 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:

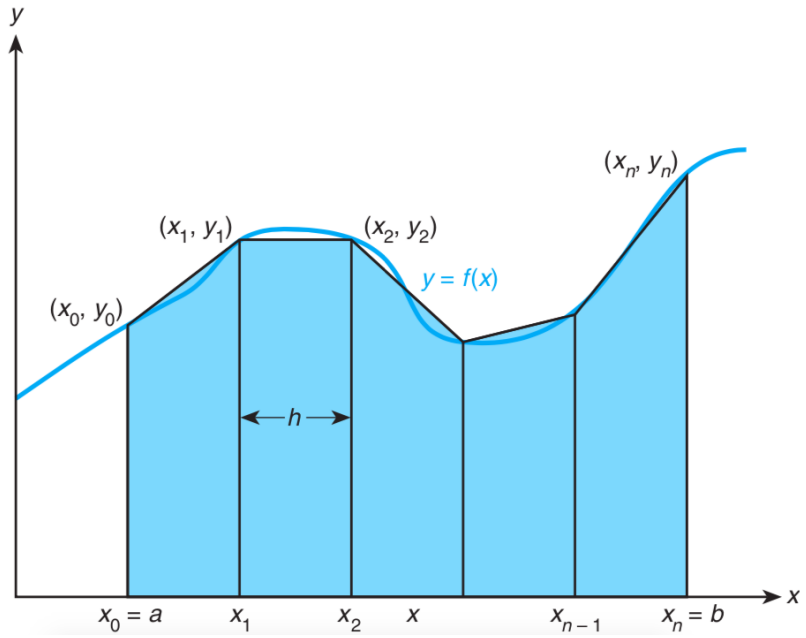


Figure 2: Approximating the Area Under a Curve with Trapezoids

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

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

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

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

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

Lab Assignment-5

5. Programming project on Pointers and Modular Programming

5.1 Write a program for an automatic teller machine that dispenses money. The user should enter the amount desired (a multiple of 10 dollars) and the machine dispenses this amount using the least number of bills. The bills dispensed are 50s, 20s, and 10s. Write a function that determines how many of each kind of bill to dispense.

5.2 Write a program to dispense change. The user enters the amount paid and the amount due. The program determines how many dollars, quarters, dimes, nickels, and pennies should be given as change. Write a function with four output parameters that determines the quantity of each kind of coin.

5.3 Determine the following information about each value in a list of positive integers.

a. Is the value a multiple of 7, 11, or 13?

b. Is the sum of the digits odd or even?

c. Is the value a prime number?

You should write a function with three type int output parameters that send back the answers to these three questions. Some sample input data might be:
104 3773 13 121 77 30751

5.4 The square root of a number N can be approximated by repeated calculation using the formula

$$NG = 0.5(LG + N/LG)$$

where NG stands for next guess and LG stands for last guess. Write a function that calculates the square root of a number using this method.

The initial guess will be the starting value of LG . The program will compute a value for NG using the formula given. The difference between NG and LG is checked to see whether these two guesses are almost identical. If they are, NG is accepted as the square root; otherwise, the next guess (NG) becomes the last guess (LG) and the process is repeated (another value is computed for NG, the difference is checked, and so on). The loop should be repeated until the difference is less than 0.005. Use an initial guess of 1.0.

Write a driver function and test your square root function for the numbers 4, 120.5, 88, 36.01, 10,000, and 0.25.

5.5 When an aircraft or an automobile is moving through the atmosphere, it must overcome a force called drag that works against the motion of the vehicle. The drag force can be expressed as

$$F = \frac{1}{2} CD \times A \times \rho \times V^2$$

where F is the force (in newtons), CD is the drag coefficient, A is the projected area of the vehicle perpendicular to the velocity vector (in m^2), is the density of the gas or fluid through which the body is traveling (kg/m^3), and V is the body's velocity. The drag coefficient CD has a complex derivation and is frequently an empirical quantity. Sometimes the drag coefficient has its own dependencies on velocities: For an automobile, the range is from approximately 0.2 (for a very streamlined vehicle) through about 0.5. For simplicity, assume a streamlined passenger vehicle is moving through air at sea level (where $\rho = 1.23 kg/m^3$). Write a program that allows a user to input A and CD interactively and calls a function to compute and return

the drag force. Your program should call the drag force function repeatedly and display a table showing the drag force for the input shape for a range of velocities from 0 m/s to 40 m/s in increments of 5 m/s.

5.6 Write a program to model a simple calculator. Each data line should consist of the next operation to be performed from the list below and the right operand. Assume the left operand is the accumulator value (initial value of 0). You need a function **scan_data** with two output parameters that returns the operator and right operand scanned from a data line. You need a function **do_next_op** that performs the required operation. **do_next_op** has two input parameters (the operator and operand) and one input/output parameter (the accumulator). The valid operators are:

+ add
− subtract
∗ multiply
/ divide
^ power (raise left operand to power of right operand)
q quit

Your calculator should display the accumulator value after each operation. A sample run follows.

+ 5.0
result so far is 5.0
^ 2
result so far is 25.0
/ 2.0
result so far is 12.5
q 0
final result is 12.5

5.7 After studying gross annual revenues of Broadway shows over a 20-year period, you model the revenue as a function of time:

$$R(t) = 203.265 \times (1.071)^t$$

where R is in millions of dollars and t is the years since 1984. Create the following C functions to implement this model:

revenue—calculates and returns R for an input parameter of t.

predict—predicts the year in which revenues (in millions) will first equal or exceed the value of the input parameter. For example, **predict(200)** would return 1984.

Write a main function that calls **predict** to determine when revenues will likely exceed $1 trillion (i.e., 1,000 million). Then create an output file that contains a table of estimated revenues (in millions of dollars) for all the years from 1984 through the year when revenues should exceed $1 trillion. Round revenue estimates to three decimal places.

5.8 Since communications channels are often noisy, numerous ways have been devised to ensure reliable data transmission. One successful method uses a checksum. A checksum for a message

can be computed by summing the integer codes of the characters in the message and finding the remainder of this sum divided by 64. The integer code for a space character is added to this result to obtain the checksum. Since this value is within the range of the displayable characters, it is displayed as a character as well. Write a program that accepts single-line messages ending with a period and displays the checksum character for each message. Your program should continue displaying check- sums until the user enters a line with only a period.

5.9 Harlan A. Brothers and John A. Knox discovered that as the value of x gets larger, the value of the expression $\left(\frac{2x+1}{2x-1}\right)^x$ gets closer and closer to e. Write a program that evaluates this expression for $x = 1, 2, 3$, and so on until the absolute difference between the expression's value and the value of e calculated by the library function **exp** is less than 0.000001. Display the value of x that causes your loop to exit along with both the final approximation of e and the value of e calculated by the **exp** function. Show 7 decimal places.

Lab Assignment-6

6. Programming project on Arrays and Strings

6.1 Write a program to take two numerical lists of the same length ended by a sentinel value and store the lists in arrays x and y, each of which has 20 elements. Let n be the actual number of data values in each list. Store the product of corresponding elements of x and y in a third array, z, also of size 20. Display the arrays x, y, and z in a three-column table. Then compute and display the square root of the sum of the items in z. Make up your own data, and be sure to test your program on at least one data set with number lists of exactly 20 items. One data set should have lists of 21 numbers, and one set should have significantly shorter lists.

6.2 A barcode scanner for Universal Product Codes (UPCs) verifies the 12-digit code scanned by comparing the code's last digit (called a check digit) to its own computation of the check digit from the first 11 digits as follows:

1. Calculate the sum of the digits in the odd-numbered positions (the first, third, ..., eleventh digits) and multiply this sum by 3.
2. Calculate the sum of the digits in the even-numbered positions (the second, fourth, ..., tenth digits) and add this to the previous result.
3. If the last digit of the result from step 2 is 0, then 0 is the check digit. Otherwise, subtract the last digit from 10 to calculate the check digit.
4. If the check digit matches the final digit of the 12-digit UPC, the UPC is assumed correct.

Write a program that prompts the user to enter the 12 digits of a barcode separated by spaces. The program should store the digits in an integer array, calculate the check digit, and compare it to the final barcode digit. If the digits match, output the barcode with the message "validated." If not, output the barcode with the message "error in barcode." Also, output with labels the results from steps 1 and 2 of the check-digit calculations. Note that the "first" digit of the barcode will be stored in element 0 of the array. Try your program on the following barcodes, three of which are valid. For the first barcode, the result from step 2 is 79 (0 + 9 + 0 + 8 + 4 + 0) * 3 + (7 + 4 + 0 + 0 + 5).

079400804501

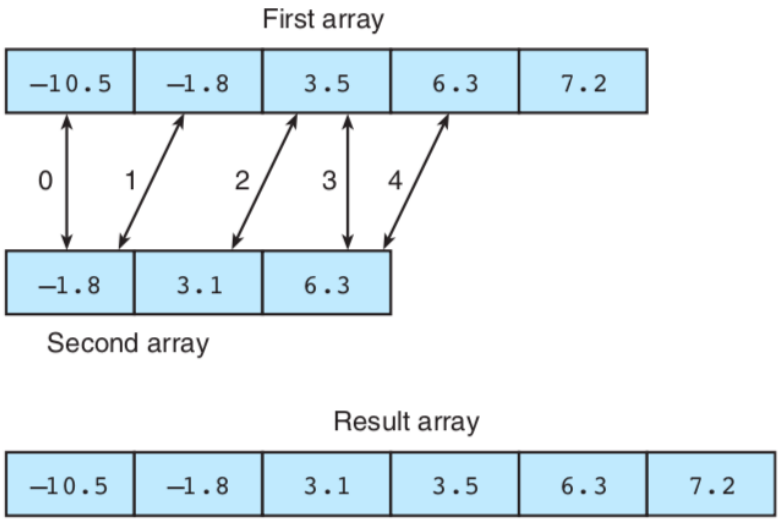
024000162860

011110856807

051000138101

6.3 Each year the Department of Traffic Accidents receives accident count reports from a number of cities and towns across the country. To summarize these reports, the department provides a frequency distribution printout that gives the number of cities reporting accident counts in the following ranges: 0–99, 100–199, 200–299, 300–399, 400–499, and 500 or above. The department needs a computer program to take the number of accidents for each reporting city or town and add one to the count for the appropriate accident range. After all the data have been processed, the resulting frequency counts are to be displayed.

6.4 Write a function that will merge the contents of two sorted (ascending order) arrays of type double values, storing the result in an array output parameter (still in ascending order). The function should not assume that both its input parameter arrays are the same length but can assume that one array does not contain two copies of the same value. The result array should also contain no duplicate values.



Hint: When one of the input arrays has been exhausted, do not forget to copy the remaining data in the other array into the result array. Test your function with cases in which (1) the first array is exhausted first, (2) the second array is exhausted first, and (3) the two arrays are exhausted at the same time (i.e., they end with the same value). Remember that the arrays input to this function must already be sorted.

6.5 The binary search algorithm that follows may be used to search an array when the elements are in order. This algorithm is analogous to the following approach for finding a name in a telephone book.

a. Open the book in the middle, and look at the middle name on the page.
b. If the middle name isn't the one you're looking for, decide whether it comes before or after the name you want.
c. Take the appropriate half of the section of the book you were looking in and repeat these steps until you land on the name.

ALGORITHM FOR BINARY SEARCH

1. Let **bottom** be the subscript of the initial array element.
2. Let **top** be the subscript of the last array element.
3. Let **found** be false.
4. Repeat as long as **bottom** isn't greater than **top** and the target has not been found
5. Let middle be the subscript of the element halfway between **bottom** and **top**.
6. if the element at middle is the target
7. Set **found** to true and **index** to middle.
   else if the element at middle is larger than the target
8. Let **top** be middle - 1.
   else
9. Let **bottom** be middle + 1.

Write and test a function **binary_srch** that implements this algorithm for an array of integers. When there is a large number of array elements, which function do you think is faster: **binary_srch** or the linear search function?

6.6 The bubble sort is another technique for sorting an array. A bubble sort compares adjacent array elements and exchanges their values if they are out of order. In this way, the smaller values "bubble" to the top of the array (toward element 0), while the larger values sink to the

bottom of the array. After the first pass of a bubble sort, the last array element is in the correct position; after the second pass the last two elements are correct, and so on. Thus, after each pass, the unsorted portion of the array contains one less element. Write and test a function that implements this sorting method.

6.7 A C program can represent a real polynomial $p(x)$ of degree n as an array of the real coefficients $a_0, a_1, ..., a_n (a_n \neq 0)$.

$$p(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n$$

Write a program that inputs a polynomial of maximum degree 8 and then evaluates the polynomial at various values of x. Include a function **get_poly** that fills the array of coefficients and sets the degree of the polynomial, and a function **eval_poly** that evaluates a polynomial at a given value of x. Use these function prototypes:

void get_poly(double coeff[ ], int* degreep);

double eval_poly(const double coeff[ ], int degree, double x);

6.8 Write and test a function **deblank** that takes a string output and a string input argument and returns a copy of the input argument with all blanks removed.

6.9 Write and test a function **hydroxide** that returns a 1 for true if its string argument ends in the substring **OH**.

Try the function hydroxide on the following data:

KOH H2O2 NaCl NaOH C9H8O4 MgOH

6.10 Write a program that takes nouns and forms their plurals on the basis of these rules:

a. If noun ends in "y", remove the "y" and add "ies".
b. If noun ends in "s", "ch", or "sh", add "es".
c. In all other cases, just add "s".

Print each noun and its plural. Try the following data:

chair dairy boss circus fly dog church clue dish

6.11 Write a program that takes data a line at a time and reverses the words of the line. For example,

Input: birds and bees

Reversed: bees and birds

The data should have one blank between each pair of words.

6.12 Write and test a function that finds and returns through an output parameter the longest common suffix of two words (e.g., the longest common suffix of "procrastination" and "destination" is "stination", of "globally" and "internally" is "ally", and of "gloves" and "dove" is the empty string).

Lab Assignment-7

7. Programming project on Recursion, Structure, Union and Dynamic Data Structures

---

7.1 A palindrome consists of a word or deblanked, unpunctuated phrase that is spelled exactly the same when the letters are reversed. Write a recursive function that returns a value of 1 if its string argument is a palindrome. Notice that in palindromes such as level, deed, sees, and Madam I'm Adam (madamimadam), the first letter matches the last, the second matches the next-to-last, and so on.

7.2 Write and test a recursive function that returns the value of the following recursive definition:

$$f(x) = 0 \quad if \quad x \leq 0$$

$$f(x) = f(x-1) + 2 \quad otherwise$$

What set of numbers is generated by this definition?

7.3 In question 6.5 of assignment 6, description of an iterative algorithm for searching for a target value in a sorted list is given. Here again is the introduction to that problem. The binary search algorithm that follows may be used to search an array when the elements are in order. This algorithm is analogous to the following approach to finding a name in a telephone book.

a. Open the book in the middle, and look at the middle name on the page.
b. If the middle name isn't the one you're looking for, decide whether it comes before or after the name you want.
c. Take the appropriate half of the section of the book you were looking in and repeat these steps until you land on the name.

ALGORITHM FOR BINARY SEARCH

1. Let **bottom** be the subscript of the initial array element.
2. Let **top** be the subscript of the last array element.
3. Let **found** be false.
4. Repeat as long as **bottom** isn't greater than **top** and the target has not been found
5. Let middle be the subscript of the element halfway between **bottom** and **top**.
6. if the element at middle is the target
7. Set **found** to true and **index** to middle.
   else if the element at middle is larger than the target
8. Let **top** be middle - 1.
   else
9. Let **bottom** be middle + 1.

Develop a recursive binary search algorithm, and write and test a function **binary_srch** that implements the algorithm for an array of integers.

7.4 Define a structure type **auto_t** to represent an automobile. Include components for the make and model (strings), the odometer reading, the manufacture and purchase dates (use another user-defined type called **date_t**), and the gas tank (use a user-defined type **tank_t** with components for tank capacity and current fuel level, giving both in gallons). Write I/O functions scan_date, scan_tank, scan_auto, print_date, print_tank, and print_auto, and also write a driver

function that repeatedly fills and displays an auto structure variable. Here is a small data set to try:

Mercury Sable 99842 1 18 2001 5 30 1991 16 12.5
Mazda Navajo 123961 2 20 1993 6 15 1993 19.3 16.7

7.5 Define a structure type **element_t** to represent one element from the periodic table of elements. Components should include the atomic number (an integer); the name, chemical symbol, and class (strings); a numeric field for the atomic weight; and a seven-element array of integers for the number of electrons in each shell. The following are the components of an **element_t** structure for sodium.

11 Sodium Na alkali_metal 22.9898 2 8 1 0 0 0 0
Define and test I/O functions **scan_element** and **print_element**.

7.6 Write a program using C for creation of singly linked list and perform the operation given below. The singly linked list as shown below:

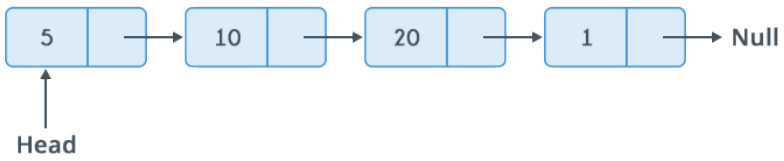


Figure 3: Singly link list

Write the **insertion** function to add a node with another data value in the above linked list. Also, create a **display** function which shows elements of linked list after insertion.