1. Write a program to swap two variables by passing the reference of these variables into a function declared as void swap(int *, int *).
2. Write a program to take numbers in array and access them through pointers. Pass data to a function and then function will sort the data of array using bubble sort.
3. Write a program which takes radius of a circle input from the user and pass it to a function called AreaPerimeter(). This function calculates the area and perimeter of a circle and return both, area and perimeter, to main().
4. An array of five elements passed through a function sum(). Calculate the sum and returns sum to main to print in the main().
5. Write a single function that receives an array of 5 integers and returns the sum, average and standard deviation of these numbers without using return statement. Call this function from main( ) and print the results in main( ).

   $$\sigma = \sqrt{\frac{\Sigma(x-\mu)^2}{N}}$$

   Where x represents each value in the population, μ is the mean value of the population, Σ is the summation (or total), and N is the number of values in the population.
6. Write a modular program that accepts at least 10 integer test scores from the user and stores them in an array. Then main should display how many perfect scores were entered (i.e., scores of 100), using a value-returning countPerfect function to help it.
7. 8. A local zoo wants to keep track of how many pounds of food each of its three monkeys eats each day during a typical week. Write a program that stores this information in a two dimensional 3 × 7 array, where each row represents a different monkey and each column represents a different day of the week.

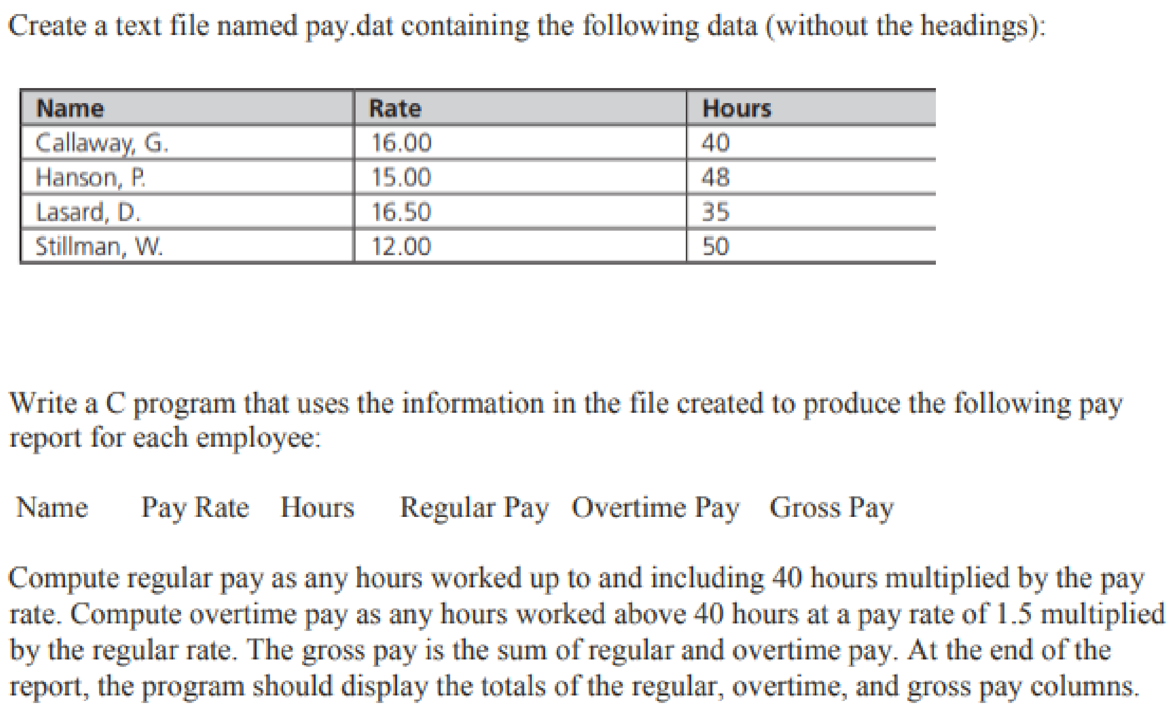
8. The program should first have the user input the data for each monkey. Then it should create a report that includes the following information:
   - Average amount of food eaten per day by the whole family of monkeys.
   - The least amount of food eaten during the week by any one monkey.
   - The greatest amount of food eaten during the week by any one monkey.

9. Write a program that can be used to gather statistical data about the number of movies college students see in a month. The program should ask the user how many students were surveyed and dynamically allocate an array of that size. The program should then allow the user to enter the number of movies each student has seen. The program should then calculate the average, median, and mode of the values entered.

10. Write a program to sort a set of names stored in an array in alphabetical order.

11. Write a program to reverse the strings.
12. Write a program in C to compute the sum of all elements in an array using pointers.
13. Write a program in C to find the factorial of a given number using pointers.
14. Write a program in C to calculate the length of the string using a pointer.

15. Write a program in C to count the number of vowels and consonants in a string using a pointer.
16. Write a program in C to print all permutations of a given string using pointers. Output should be like The permutations of the string are: abcd abdc acbd acdb adcb adbc bacd badc bcad bcda bdca bdac cbad cbda cabd cad b cdab cdba db ca dbac dcba dcab dacb dabc
17. Write a function named Sum_Num(float* , int) which receives a float array and its size and returns the sum of numbers in the array. Call this function from main. Use appropriate parameters and return type.
18. Write a program to find the max of an integral data set. The program will ask the user to input the number of data values in the set and each value. The program prints on screen a pointer that points to the max value.
19. Write a program that declares three one-dimensional arrays named price, quantity, and amount. Each array should be declared in main() and be capable of holding 10 double-precision numbers. The numbers to be stored in price are 10.62, 14.89, 13.21, 16.55, 18.62, 9.47, 6.58, 18.32, 12.15, and 3.98. The numbers to be stored in quantity are 4, 8.5, 6, 7.35, 9, 15.3, 3, 5.4, 2.9, and 4.8. Have your program pass these three arrays to a function called extend(), which calculates the elements in the amount array as the product of the equivalent elements in the price and quantity arrays: for example, amount[1]=price[1] *quantity[1]
20. Write a C program that asks for two lowercase characters. Pass the two entered characters, using pointers, to a function named capit(). The capit() function should capitalize the two letters and return the capitalized values to the calling function through its pointer arguments. The calling function should then display all four letters.
21.



Create a text file named pay.dat containing the following data (without the headings):

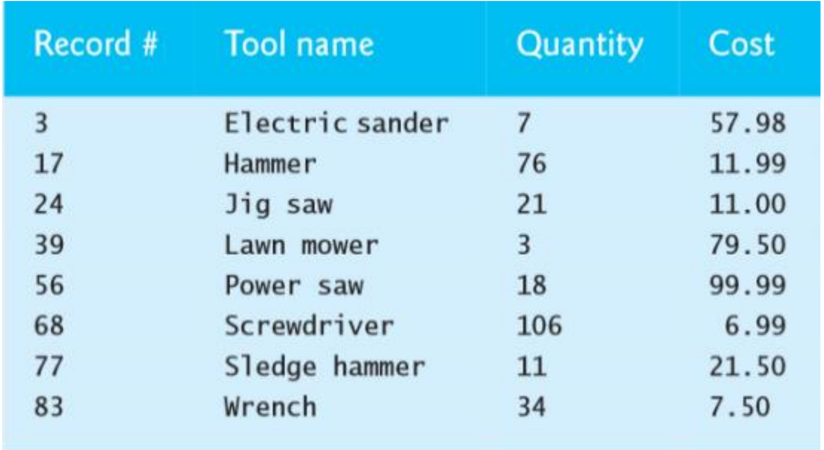
Name	Rate	Hours
Callaway, G.	16.00	40
Hanson, P.	15.00	48
Lasard, D.	16.50	35
Stillman, W.	12.00	50

Write a C program that uses the information in the file created to produce the following pay report for each employee:

```
Name      Pay Rate    Hours     Regular Pay   Overtime Pay    Gross Pay
```

Compute regular pay as any hours worked up to and including 40 hours multiplied by the pay rate. Compute overtime pay as any hours worked above 40 hours at a pay rate of 1.5 multiplied by the regular rate. The gross pay is the sum of regular and overtime pay. At the end of the report, the program should display the totals of the regular, overtime, and gross pay columns.

22. You're the owner of a hardware store and need to keep an inventory that can tell you what tools you have, how many you have and the cost of each one. Write a program that initializes the file "hardware.txt" to 8 empty records, lets you input the data concerning each tool, enables you to list all your tools, lets you delete a record for a tool that you no longer have and lets you update any information in the file. The tool identification number should be the record number. Use the following information to start your file:



Record #	Tool name	Quantity	Cost
3	Electric sander	7	57.98
17	Hammer	76	11.99
24	Jig saw	21	11.00
39	Lawn mower	3	79.50
56	Power saw	18	99.99
68	Screwdriver	106	6.99
77	Sledge hammer	11	21.50
83	Wrench	34	7.50

23. A small airline has just purchased a computer for its new automated reservations system. The president has asked you to program the new system. You'll write a program to assign seats on each flight of the airline's only plane (capacity: 10 seats). Your program should display the following menu of alternatives: Please type 1 for "first class" Please type 2 for "economy" If the person types 1, then your program should assign a seat in the first class section (seats 1– 5). If the person types 2, then your program should assign a seat in the economy section (seats 6– 10). Your program should then print a boarding pass indicating the person's seat number and whether it's in the first class or economy section of the plane. Use a single-subscripted array to represent the seating chart of the plane. Initialize all the elements of the array to 0 to indicate that all seats are empty. As each seat is assigned, set the corresponding element of the array to 1 to indicate that the seat is no longer available. Your program should, of course, never assign a seat that has already been assigned. When the first class section is full, your program should ask the person if it's acceptable to be placed in the economy section (and vice versa). If yes, then make the appropriate seat assignment. If no, then print the message "Next flight leaves in 3 hours."

24. create a text-based, menu-driven program that allows the user to choose whether to add, subtract, multiply or divide two numbers. The program should then input two double values from the user, perform the appropriate calculation and display the result. Use an array of function pointers in which each pointer represents a function that returns void and receives two double parameters. The corresponding functions should each display messages indicating which calculation was performed, the values of the parameters and the result of the calculation.

25. Write a program that inputs a line of text and a search string from the keyboard. Using function strstr, locate the first occurrence of the search string in the line of text, and assign the location to variable searchPtr of type char *. If the search string is found, print the

remainder of the line of text beginning with the search string. Then, use strstr again to locate the next occurrence of the search string in the line of text. If a second occurrence is found, print the remainder of the line of text beginning with the second occurrence. [Hint: The second call to strstr should contain searchPtr + 1 as its first argument.]

26. Write a program that inputs several lines of text and a search character and uses function strchr to determine the total occurrences of the character in the lines of text

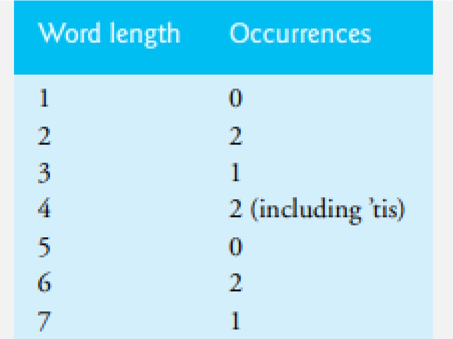
27. Write a program that inputs several lines of text and uses function strchr to determine the total occurrences of each letter of the alphabet in the lines of text. Uppercase and lowercase letters should be counted together. Store the totals for each letter in an array and print the values in tabular format after the totals have been determined.

28. Dates are commonly printed in several different formats in business correspondence. Two of the more common formats are 07/21/2003 and July 21, 2003 Write a program that reads a date in the first format and prints it in the second format

29. The availability of computers with string-manipulation capabilities has resulted in some rather interesting approaches to analyzing the writings of great authors. Much attention has been focused on whether William Shakespeare ever lived. Some scholars find substantial evidence that Christopher Marlowe actually penned the masterpieces attributed to Shakespeare. Researchers have used computers to find similarities in the writings of these two authors. This exercise examines three methods for analyzing texts with a computer.

a) Write a program that reads several lines of text and prints a table indicating the number of occurrences of each letter of the alphabet in the text. For example, the phrase To be, or not to be: that is the question: contains one “a,” two “b’s,” no “c’s,” and so on.

b) Write a program that reads several lines of text and prints a table indicating the number of one-letter words, two-letter words, three-letter words, and so on, appearing in the text. For example, the phrase Whether 'tis nobler in the mind to suffer contain



Word length	Occurrences
1	0
2	2
3	1
4	2 (including 'tis)
5	0
6	2
7	1