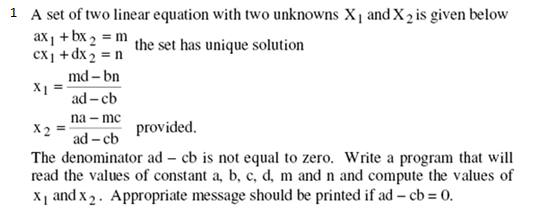
**PSG COLLEGE OF TECHNOLOGY, COIMBATORE – 641 004**

**DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCES**

**I M.Sc (SS/TCS) - C Programming Lab – Work Sheet 2**



2. Write a C program that outputs the day of the week given a date expressed as j (day) m (month) a (year). You will use the following formula:

If m is 1 m1 = 13 and a1 = a-1

If m is 2 m1 = 14 and a1 = a-1

Otherwise m1 is m and a1 is a

and with ns being the first two digits (from one’s place) of a1 and as the last two digits of a1

The day of the week will then be given by the modulo of f and 7 (0 is Monday, 1 is Tuesday etc.)

3. Write a program that takes as input a number *y* denoting the year and a number d, and prints the date which is the dth day of the year *y*. Suppose y is given as 2011 and d as 62, then your program should print 3/3/2011 Thursday.

4. Suppose we wish to write a program that plays cards. The first step in such a program would be to represent cards using numbers. In a standard deck, there are 52 cards, 13 of each suite. There are 4 suites : spades, hearts, diamonds and clubs. The 13 cards of each suit have the denomination 2,3,4,5,6,7,8,9,10,J,Q,K,A, where the last 4 respectively are short for jack, queen, king, and ace. It is natural to assign the numbers 3,2,1,0 to the suites respectively. The denominations 2-10 are assigned numbers same as the denomination, whereas the jack, queen, king, ace are respectively assigned the numbers 11,12,13, and 1 respectively. The number assigned to a card of suite s and denomination d is then 13s+d. Thus, the club ace has the smallest denomination, 1 and the spade king the highest 52. Write a program which takes a number and prints out what card it is. So given, 20 your program should print “7 of diamonds”, of given 51, it should print “queen of spades”.

5. Write a program that reads in three characters. If the three characters consists of two digits with a ‘.’ Between them, then your program should print the square of the decimal number represented by the character. Otherwise your program should print a message saying that the input given is invalid.

6. Create a program that accepts a letter from the user and returns the equivalent digit on the dial or keypad of a telephone. Telephone letters match digits as follows:

ABC=2 DEF=3 GHI=4 JKL=5

MNO=6 PRS=7 TUV=8 WXY=9

Notice that the letter Q is not used and does not need to be accounted for (can be ignored). The function should work with upper and lower case letters.

7. Write a else-if ladder or switch statement that assigns the value of 1 to quality Points if the value in grade is 'D', assigns the value of 2 to quality Points if the value in grade is 'C', assigns the value of 3 to quality Points if the value in grade is 'B', and assigns the value of 4 to quality Points if the value in grade is 'A'. If none of the above applies, quality Points should be assigned the value of 0.

8. Create a program that reads an integer value. Assume it is the number of the month and then display the name of the month.

9. (Enforcing Privacy with Cryptography) The explosive growth of Internet communications and data storage on Internet-connected computers has greatly increased privacy concerns. The field of cryptography is concerned with coding data to make it difficult for unauthorized users to read. In this exercise you’ll investigate a simple scheme for encrypting and decrypting data. A company that wants to send data over the Internet has asked you to write a program that will encrypt it so that it may be transmitted more securely. All the data is transmitted as four-digit integers. Your application should read a four-digit integer entered by the user and encrypt it as follows: Replace each digit with the result of adding 7 to the digit and getting the remainder after dividing the new value by 10. Then swap the first digit with the third, and swap the second digit with the fourth. Then print the encrypted integer. Write a separate application that inputs an encrypted four-digit integer and decrypts it (by reversing the encryption scheme) to form the original number.

10. 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

11. 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.

12. 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 ≤n < 5.5 Some damage

5.5 ≤ n < 6.5 Serious damage: walls may crack or fall

6.5 ≤ 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.

13. 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 and height in inches, use this formula (rounded to tenths):



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

14. 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

15. 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. 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.

16. 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:

*side*1 = *m*2 - *n*2

*side*2 = 2*mn*

*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.

17. Calculate the value of π from the infinite series



Print a table that shows the value of π approximated by one term of this series, by two terms, by three terms, and so on. How many terms of this series do you have to use before you first get 3.14? 3.141? 3.1415? 3.14159?

18. Write a program that prints a table of the binary, octal and hexadecimal equivalents of the decimal numbers in the range 1 through 256.

19. The *factorial* function is used frequently in probability problems. The factorial of a positive integer *n* (written *n*! and pronounced “*n* factorial”) is equal to the product of the positive integers from 1 to *n*. Write a program that evaluates the factorials of the integers from 1 to 5. Print the results in tabular format. What difficulty might prevent you from calculating the factorial of 20?

20. Write a program that sums a sequence of integers. Assume that the first integer read with scanf specifies the number of values remaining to be entered. Your program should read only one value each time scanf is executed. A typical input sequence might be

5 100 200 300 400 500

where the 5 indicates that the subsequent five values are to be summed.

21. Write a program that calculates and prints the average of several integers. Assume the last value read with scanf is the sentinel -999. A typical input sequence might be

10 8 11 7 9 -999

indicating that the average of all the values preceding 9999 is to be calculated.

22. Write a program that finds the smallest of several integers. Assume that the first value read specifies the number of values remaining.

23. Write a program that calculates and prints the sum of the even integers from 2 to 30.

24. Write a program to find the sum of the digits of the given integer number.

25. 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

26. 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 following formula.



This payment must be rounded to the nearest rupees. 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 Rs.1000 loan borrowed at a 9% annual interest rate and paid back over 6 months.

Principal Rs.1000 Payment Rs.171.07

Annual interest 9.0% Term 6 months

**Payment Interest Principal Principal**

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

27. Write a program that will find both the range of values in the data collection and the standard deviation of the data collection. Get the value of *N* before scanning each value in the collection of *N* numbers. 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

**

28. 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

Display the average temperature (a real number) at the end of the run.

29. 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.

30. 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.)