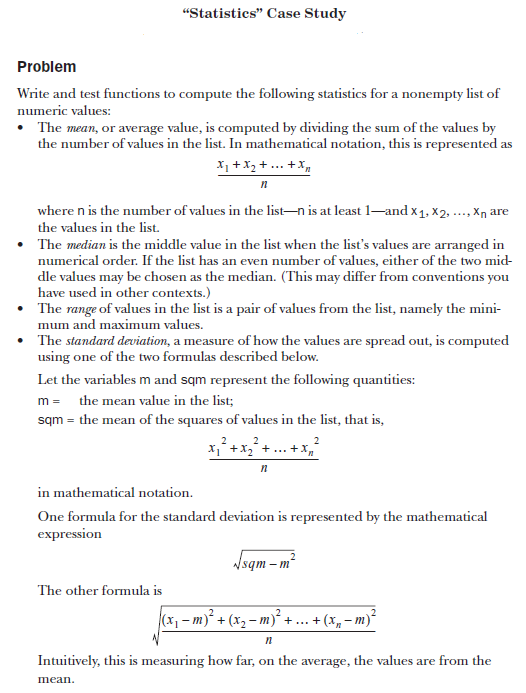
**PSG COLLEGE OF TECHNOLOGY, COIMBATORE – 641 004**

**DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCES**

**Worksheet -6**

**Class : M.Sc (SS/TCS) Semester:1 Course :C Programming Lab**

1.



2. A famous method for producing a list of prime numbers is called the Sieve of Eratosthenes. You start with a list of all the numbers from 2 through N, and cross out all the multiples of 2 (except for 2). You then find the next smallest number that has not been crossed out (3), and cross out all the multiples of this, then the multiples of the next smallest uncrossed number (5) ... and so on. The numbers that are never crossed out are the primes. This can be implemented by using addition only, so is spectacularly fast. Develop a program that will implement this algorithm and use it to find the primes less than some upper limit read in as data.

Hint: use an array

int sieve[MAX] ;

All elements start with values of true; crossing elements out is achieved by assigning a value of false to the appropriate element.

3. Julian Day

Julian day is the number of days inclusive for a day counting from New Year’s day inclusive. January 20 is Julian day 20, April 15 is Julian day 31 + 28 + 31 + 15 = 105 for an ordinary year and April 15 is Julian day 31 + 29 + 31 + 15 = 106 for a leap year.

You can compute the Julian day to figure out how many days had passed since new year’s day in this year. You can also compute Julian days since the new year day a few years ago. The only thing that can complicate things is the leap year (and have a function intisLeapYear (int) that takes an integer as input and compute and return if a year is leap year or not.

Logic to Compute Julian Day:

Consider the simple case that this year is an ordinary year (there are 28 days in February).

Code to calculate Julian days using array:

For an ordinary year, what we need is just;

intmonthdays [13]= {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 31, 30};

// number of days in each month

for (int j = 1; j < m; j +=)

jd += monthdays [j]; // add number of days in the month before month m

jd += d; // add the number of days in d.

You see, the number of code is 4 lines compared with 40 lines for an ordinary year.

Write a C program for Julian days (and inverse)

Step 1. Prompt for year y, month m, and day d.

Step 2. Check if year y is leap year

Step 3. Compute Julian day jd with month m and day d in year y.

Step 4. Make sure your code handles the invalid m and invalid d.

Step 5. Test with these data triples (m, d, y) = (6, 31, 2004), = (0,20, 2005), = (10, 31, 2006), = (2, 29, 2006), = (2, 28, 2006). The first triple has invalid day, the second triple has invalid month, the third triple has good data. How about the 4th and the 5th triple?

Step 6. Do the reverse. Assume we know today is Julian day jd of the year y, compute m and d of the year. For example, if today is Julian day 183 (the middle of the year), which month and which day is that?

4. Write a C program that reads N positive integers one at a time and check whether they have been input in sorted order (ascending).

5. You are given an array A of n integers. It is given that the elements of A satisfy the following inequalities A[0]< A[1]<···< A[m−1]< A[m]> A[m+ 1]> A[m+ 2]>···> A[n−1]for some (unknown) index m in the range 1<= m<=n−2. Let us call such an array a hill-valued array. The sequence A[0], A[1], . . . , A[m−1], A[m]is called the ascending part of the hill, and the remaining part A[m], A[m+ 1], . . . , A[n−1]is called the descending part of the hill. The element A[m]is the peak of the hill and is the largest element in the array.Your task is to locate the peak (that is,A[m]) in the hill-valued arrayA and print m and A[m].

**6. Given an array A[] and a number x, check for pair in A[] with sum as x**

Write a C program that, given an array A[] of n numbers and another number x, determines whether or not there exist two elements in A whose sum is exactly x.

Algorithm:

hasArrayTwoCandidates (A[], ar\_size, sum)

1) Sort the array in non-decreasing order.

2) Initialize two index variables to find the candidate

elements in the sorted array.

(a) Initialize first to the leftmost index: l = 0

(b) Initialize second the rightmost index: r = ar\_size-1

3) Loop while l < r.

(a) If (A[l] + A[r] == sum) then return 1

(b) Else if( A[l] + A[r] < sum ) then l++

(c) Else r--

4) No candidates in whole array - return 0

Example:  
Let Array be {1, 4, 45, 6, 10, -8} and sum to find be 16

Sort the array  
A = {-8, 1, 4, 6, 10, 45}

Initialize l = 0, r = 5  
A[l] + A[r] ( -8 + 45) > 16 => decrement r. Now r = 10  
A[l] + A[r] ( -8 + 10) < 2 => increment l. Now l = 1  
A[l] + A[r] ( 1 + 10) < 16 => increment l. Now l = 2  
A[l] + A[r] ( 4 + 10) < 14 => increment l. Now l = 3  
A[l] + A[r] ( 6 + 10) == 16 => Found candidates (return 1)

Note: If there are more than one pair having the given sum then this algorithm reports only one. Can be easily extended for this though

# 7. Find the smallest positive number missing from an unsorted array

You are given an unsorted array with both positive and negative elements. You have to find the smallest positive number missing from the array.

Examples

Input: {2, 3, 7, 6, 8, -1, -10, 15}

Output: 1

Input: { 2, 3, -7, 6, 8, 1, -10, 15 }

Output: 4

Input: {1, 1, 0, -1, -2}

Output: 2

# 8. Given an array of size n and a number k, find all elements that appear more than n/k times

Given an array of size n, find all elements in array that appear more than n/k times. For example, if the input arrays is {3, 1, 2, 2, 1, 2, 3, 3} and k is 4, then the output should be [2, 3]. Note that size of array is 8 (or n = 8), so we need to find all elements that appear more than 2 (or 8/4) times. There are two elements that appear more than two times, 2 and 3.

A **simple method** is to pick all elements one by one. For every picked element, count its occurrences by traversing the array, if count becomes more than n/k, then print the element.

A better solution is to **use sorting**. First, sort all elements. Once the array is sorted, we can find all required elements in a linear scan of array.

9. Write a C function to remove duplicates from an ordered array. For example, if input array contains 10,10,10,30,40,40,50,80,80,100 then output should be 10,30,40,50,80,100.

10. Write a program which will arrange the positive and negative numbers in a one-dimensional array in such a way that all positive numbers should come first and then all the negative numbers will come without changing original sequence of the numbers.

Example:

Original array contains: 10,-15,1,3,-2,0,-2,-3,2,-9

Modified array: 10,1,3,0,2,-15,-2,-2,-3,-9

11. Write a program to remove all articles from an input string.

For example Input : The Dhillon Theatre is now the Fun Republic.

Output : Dhillon atre is now Fun Republic.

12. Write a program to replace ‘a’ with ‘b’, ‘b’ with ‘c’,….,’z’ with ‘a’ and similarly for ‘A’ with ‘B’,’B’ with ‘C’, …., ‘Z’ with ‘A’ in the given string.

13. Given an int array of length N, develop code that stores in the i-th element the value of

1 + 2 + ... i for all N values of i.

14. Develop code that will cycle all elements of an array one position to the left. For example, if the array originally contained

-1 2 -4 3 7 6 102 41

it should end up containing

2 -4 3 7 6 102 41 -1

15. Develop code that will cycle all elements of an array of N elements by M positions to the left. What action should you take if M >= N ?

16. Develop code that will find the largest and smallest elements in an array of known length.

17. Develop code that will determine the maximum value of those held in an array, and the number of times that maximum occurs.

18. Develop code that will reverse the order of elements in an array of N elements. For example, if the array originally contained

-1 2 -4 3 7 6 102 41

it should end up containing

41 102 6 7 3 -4 2 -1

Ensure that your code behaves correctly for odd and even values of N, and also when N has the value 1.

19. Develop a program that will read a list of integer numbers and tell you how many times each of the digits 0 ... 9 appeared in each number, and how many times each of these digits appeared in all the numbers together.

20. Develop a program that will read a list of student marks in the range 0 ... 100 and tell you how many students scored in each range of 10 - how many scored 0 - 9, how many 10 -19, 20 - 29 ... and so on.

21. Develop a program that will use a random number generator to "throw" two dice on each turn, and then add the two dice together. The total must be in the range 2 to 12. Throw the dice 3600 times, and print a frequency table showing how many times each of the totals 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 occurred.

22. A rather ridiculous computer "game" is proposed as follows: The unsuspecting player is told that the computer has picked a number between 1 and 10, and she is invited to guess it in as few moves as possible. The computer actually cheats - the user is allowed to make 9 guesses, all of which are scored as "wrong", after which the computer displays a suitable message and discloses the number. The game is given to two programmers to develop. Jacob tries to be clever, and simply to add the nine numbers together. Thabo decides to develop it by using an array to keep track of the poor player's progress. Show how each of these ideas could be developed into separate, simple, complete short programs. Assume that the human player is honest and accurate (so do not bother to check that a human guess is "out of range" - assume that a nine different numbers in the range 1 through 10 will be supplied each time). What would be the effect of meeting a perverse player who chose a particular number more than once?

23. Develop and test a **function** that will return the day within the year for a given date

For example, February 5, 2009 was the 31 + 5 = 36th day of the year, while May 19, 1981 was the 31 + 28 + 31 +30 + 19 = 139th day of that year. Hint: set up an array that stores the day of the year for each of the months in a normal year to speed up the calculations. It may interest you to know that there is a clever algorithm. (Implement this also)

int dayNumber = (month - 1) \* 31 + day;

if (month > 2) {

dayNumber = dayNumber - ((4 \* month + 23) / 10);

if (isLeapYear(year)) dayNumber++;

}

Print dayNumber;

24. Develop and **test a function** that will return the score for a word in the well-known game of Scrabble, assuming that none of the letters are placed on a square that doubles or trebles any points.

The tiles in Scrabble earn basic points as follows - Q and Z score 10, J and X score 8; K scores 5; F, H, V, W and Y score 4; B, C, M and P score 3; D and G score 2; other letters score 1; blank scores 0. Hint: set up an array of 26 values to contain the scores for each letter to speed up the calculation, and ignore differences in upper and lower case.

25. Develop an application that will read a piece of text and translate it into Morse code. The code is defined by the sequences of dots and dashes given in the table below. Ignore variation in upper and lower case, separate characters by one space and words by ends of lines and simply replace characters not in the table by spaces.

For example, the frantic plea

Can you help me?

should produce the output

-.-. .- -.

-.-- --- ..-

.... . .-.. .--.

-- . ..--..

Also develop an application that will translate a message in Morse code back to English. Hint: as in the previous two exercises, use of arrays to provide lookup tables will make the exercise easier.

A .- F ..-. K -.- P .--. U ..- Z --.. 3 ...-- 8 ---..

B -... G --. L .-.. Q --.- V ...- 4 ....- 9 ----.

C -.-. H .... M -- R .-. W .-- 0 ----- 5 ..... . .-.-.-

D -.. I .. N -. S ... X -..- 1 .---- 6 -.... , --..--

E . J .--- O --- T - Y -.-- 2 ..--- 7 --... ? ..--..

26. Write a complete working program that will ask for a person’s name and his/her game score. Then it will ask for a second person’s name and score. The program will print the winner’s name and also print by how many points that person won.

27. Julian Day

Julian day is the number of days inclusive for a day counting from New Year’s day inclusive. January 20 is Julian day 20, April 15 is Julian day 31 + 28 + 31 + 15 = 105 for an ordinary year and April 15 is Julian day 31 + 29 + 31 + 15 = 106 for a leap year.

You can compute the Julian day to figure out how many days had passed since new year’s day in this year. You can also compute Julian days since the new year day a few years ago. The only thing that can complicate things is the leap year (and have a function int isLeapYear (int) that takes an integer as input and compute and return if a year is leap year or not.

Logic to Compute Julian Day:

Consider the simple case that this year is an ordinary year (there are 28 days in February).

Code to calculate Julian days using array:

For an ordinary year, what we need is just;

int monthdays [13]= {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 31, 30};

// number of days in each month

for (int j = 1; j < m; j +=)

jd += monthdays [j]; // add number of days in the month before month m

jd += d; // add the number of days in d.

You see, the number of code is 4 lines compared with 40 lines for an ordinary year.

Write a C program for Julian days (and inverse)

Step 1. Prompt for year y, month m, and day d.

Step 2. Check if year y is leap year

Step 3. Compute Julian day jd with month m and day d in year y.

Step 4. Make sure your code handles the invalid m and invalid d.

Step 5. Test with these data triples (m, d, y) = (6, 31, 2004), = (0,20, 2005), = (10, 31, 2006), = (2, 29, 2006), = (2, 28, 2006). The first triple has invalid day, the second triple has invalid month, the third triple has good data. How about the 4th and the 5th triple?

Step 6. Do the reverse. Assume we know today is Julian day jd of the year y, compute m and d of the year. For example, if today is Julian day 183 (the middle of the year), which month and which day is that?

28. Top Batsmen

Given an array Score[11] of eleven elements and an integer K, find the number of ways in which a subset of exactly K elements can be selected such that the sum of its elements is the maximum possible. It’s easy to see that to get the maximum sum, we need to pick the elements in non-increasing order. We start picking elements and suppose there are M players with score S and we need to still pick K more players. If K > M, then all the players having score S should be taken, otherwise K ≤ M and we need to select K players from these M players. This is the only choice we have to make. All players having score greater than S must be taken. Players having score less than S must not be taken, as it only decreases the total sum. So the number of ways is binomial(M,K).

Alternate Solution : As the number of players is only 11, we can try all possible K element subsets and find the sum of scores. We maintain the maxSum and count ( number of subsets so far with sum = maxSum ). If the current\_sum is greater than maxSum, update the maxSum = current\_sum and count = 1, else if its equal update count++;

29. Write a C program that reverses the order of an array that contains 100 random numbers. For example, array [0] becomes array [99], array [1] becomes array [98], and so on.

30. Write a C program that accumulates an array of temperature values and calculate the min temperature, maximum temperature and the average temperature.

31 Write a C program that simulates the roll of a die with possible values 1 through 6 and records the number of times that the die rolls each value. The input is the number of times to roll the die, and the outputs include the number of times the die falls on each possible value.

32. Write a C program that generates a 1D array and then multiplies pairs of elements together, starting with elements 0 and 1, and returns the resulting array. For example, the input array with values {1,23,10,5,7,11} results in the output array {23,50,77}.

33. Write a C program to display a question and get the answer for the question. Display “Good”, if the answer is correct and display “Wrong” if the answer is wrong.

34. Write a program to read a string and rewrite it in the alphabetical order.

35. Given a string char s[] = “123456789”, write a program that displays the following.

1

2 3 2

3 4 5 4 3

4 5 6 7 6 5 4

5 6 7 8 9 8 7 6 5

36. An election is contested by 5 candidates. The candidates are numbered 1 to 5 and the voting is done by marking the candidate number on the ballot paper. Write a program to read the ballots and count the votes cast for each candidate using an array variable count. In case, a number read is outside the range 1 to 5, the ballot should be considered as a “spoilt ballot” and the program should also count the number of spoilt ballots.

37. The following set of numbers is popularly known as Pascal’s triangle.

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

-

-

If we denote rows by i and columns by j, then any element (except boundary element) in the triangle is given by Pij = P i-1, j-1 + P i-1, j

38. Given that the contents of the following two arrays are both in ascending order,

const int MAX\_ITEMS=100;

double a [MAX\_ITEMS], b [MAX\_ITEMS];

and given that the variables *aCount* and *bCount* contain the number of elements in the arrays *a* and *b* are in use, merge the values in arrays *a* and *b* in to the following array *c* so that the values recorded in array *c* are preserved in ascending order.

double c[MAX\_ITEMS];

39. Write a program to read a string using pointer then do the following manipulations using pointers,

Find length of the string

Copy the string into another one

Reverse the string

Read one more string and then compare the new string with the old string.

40.      +-------------------------+  
                  ¦ 34 ¦ 21 ¦ 32 ¦ 41 ¦ 25  ¦  
                  +----+----+----+----+-----¦  
                  ¦ 14 ¦ 42 ¦ 43 ¦ 14 ¦ 31  ¦  
                  +----+----+----+----+-----¦  
                  ¦ 54 ¦ 45 ¦ 52 ¦ 42 ¦ 23  ¦  
                  +----+----+----+----+-----¦  
                  ¦ 33 ¦ 15 ¦ 51 ¦ 31 ¦ 35  ¦  
                  +----+----+----+----+-----¦  
                  ¦ 21 ¦ 52 ¦ 33 ¦ 13 ¦ 23  ¦  
                  +-------------------------+

Do you like treasure hunts? In this problem you are to write a program to explore the above array for a treasure. The values in the array are clues. Each cell contains an integer between 11 and 55; for each value the ten's digit represents the row number and the unit's digit represents the column number of the cell containing the next clue. Starting in the upper left corner (at 1,1), use the clues to guide your search of the array. (The first three clues are 11, 34, 42). The treasure is a cell whose value is the same as its coordinates. Your program must first read in the treasure map data into a 5 by 5 array. Your program should output the cells it visits during its search, and a message indicating where you found the treasure.

41. Write a program to search for the "saddle points" in a 5 by 5 array of integers. A saddle point is a cell whose value is greater than or equal to any in its row, and less than or equal to any in its column. There may be more than one saddle point in the array. Print out the coordinates of any saddle points your program finds. Print out "No saddle points" if there are none.

42. The results from the mayor's race have been reported by each precinct as follow

          Candidate  Candidate  Candidate  Candidate

Precinct      A          B          C          D  
   1         192        48         206        37  
   2           147        90         312        21  
   3           186        12         121        38  
   4         114        21         408        39  
   5         267        13         382        29

Write a program to do the following:  
a. Read the raw vote totals   
b. Display the table with appropriate headings for the rows and columns.  
c. Compute and display the total number of votes received by each candidate and  
 the percent of the total votes cast.  
d. If any one candidate received over 50% of the votes, the program should print a message declaring that candidate the winner.  
  
e. If no candidate received 50% of the votes, the program should print a message  
 declaring a run-off between the two candidates receiving the highest number of  
 votes; the two candidates should be identified by their letter names.

43. A bowling match consists of ten frames. Each frame except for the tenth consists of one or two balls, or attempts to knock down the ten pins at the end of the alley. Doing so on the first ball of the frame is called a strike, and the second ball of the frame is not rolled. Knocking down all ten pins with both balls (having left some up with the first ball) is called a spare. If both attempts to knock down the pins leave some standing, the frame is called an open frame. A spare in the tenth frame gives the bowler one extra ball; a strike in the tenth gives him or her two extra balls. A bowling score is computed as follows. A strike counts as 10 points plus the sum of the next two balls. A spare counts as 10 points plus the next ball. Any other balls merely count as themselves, as do any bonus balls rolled as a result of a strike or a spare in the tenth frame. Suppose for example that the sequence of balls was

9 1   0 10   10   10   6 2   7 3   8 2   10   9 0   9 1   10

The score for the ten frames would be

Frame   score  
-----   -----   
 1       10   
 2       30   
 3       56   
 4       74   
 5       82   
 6      100   
 7      120   
 8      139   
 9      148   
 10     168

Write a program to accept from standard input the scores for a sequence of balls and output the scores for the ten frames. There may be multiple lines of input, where each input line will be the scores for one player. The scores will be separated by one or more blanks. You may assume that the number of scores on the line is valid.

44. The following set of numbers is popularly known as Pascal’s triangle.

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

-

-

If we denote rows by i and columns by j, then any element (except boundary element) in the triangle is given by Pij = P i-1, j-1 + P i-1, j

45. Define 2 dimensional array A, B, Sum, Diff, Mult. Store MxNarbitrary numbers in A and B and Do the following:

a) Calculate sum of A and B where sum(i,j)=a(i,j)+b(i,j)

b) Calculate difference A and B and store in diff where diff(i,j)=a(i,j)- b(i,j)

c) Calculate product of two arrays A and B and store in mult where mult(i,j)= summation of a(i,k)\*b(k,j) over k where k=1 to m.

Print the result in a tabular form

46. Define 2 dimensional array a (3,3), b(3,3),sum(3,3),diff(3,3),mult(3,3). Store 9 arbitrary numbers in a(3,3) and 9 arbitrary numbers in b(3,3). Do the following:

a) Calculate sum of a(3,3) and b(3,3) and store in sum(3,3) where sum(i,j)=a(i,j)+b(i,j)

b) Calculate difference of a(3,3) and b(3,3) and store in diff(3,3) where diff(i,j)=a(i,j)- b(i,j)

c) Calculate product of two arrays a(3,3) and b(3,3) and store in mult(3,3) where mult(i,j)= summation of a(i,k)\*b(k,j) over k where k=1 to 3.

Print the result in a tabular form