



OOP Assignment – 1

(The program should be written in C++)

- ✓ 1. WAP (Write a program) to print out all Armstrong numbers between 1 and 500. If sum of cubes of each digit of the number is equal to the number itself, then the number is called an Armstrong number. For example, $153 = (1 * 1 * 1) + (5 * 5 * 5) + (3 * 3 * 3)$
- ✓ 2. WAP for a matchstick game being played between the computer and a user. Your program should ensure that the computer always wins. Rules for the game are as follows:
 - a. There are 21 matchsticks.
 - b. The computer asks the player to pick 1, 2, 3, or 4 matchsticks.
 - c. After the person picks, the computer does its picking.
 - d. Whoever is forced to pick up the last matchstick loses the game.
- ✓ 3. WAP to fill the entire screen with diamond and heart alternatively. The ASCII value for heart is 3 and that of diamond is 4.
- ✓ 4. The natural logarithm can be approximated by the following series. If x is input through the keyboard, WAP to calculate the sum of first n terms (input through the keyboard) of this series.

$$\frac{1}{2} \left(\frac{x-1}{x} \right) + \frac{1}{2} \left(\frac{x-1}{x} \right)^2 + \frac{1}{2} \left(\frac{x-1}{x} \right)^3 + \frac{1}{2} \left(\frac{x-1}{x} \right)^4 + \dots$$

- ✓ 5. Write a menu driven program which has following options:
 - a. Factorial of a number.
 - b. Prime or not
 - c. Odd or even
 - d. Exit
- ✓ 6. WAP to find the norm of a matrix. The norm is defined as the square root of the sum of squares of all elements in the matrix.
- ✓ 7. The X and Y coordinates of 10 different points are entered through the keyboard. WAP to find the distance of last point from the first point (sum of distance between consecutive points).\
- ✓ 8. WARP (Write a recursive program) to evaluate $f(x) = x + x^3/3! + x^5/5! + \dots + x^n/n!$ where n is the input to be supplied by user.
- ✓ 9. WAP to sort a set of names stored in an array in alphabetical order.
- ✓ 10. WAP to remove characters in string except alphabets
- ✓ 11. WAP that receives the month and year from the keyboard as integers and prints the calendar in the following. Note that according to the Gregorian calendar 01/01/1900 was Monday. With this as the base the calendar should be generated.

September 2004						
Mon	Tue	Wed	Thu	Fri	Sat	Sun
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			





12. Write a menu driven program that depicts the working of a library. The menu options should be:
- Add book information
 - Display book information
 - List all books of given author
 - List the title of specified book
 - List the count of books in the library
 - List the books in the order of accession number
 - Exit

Create a structure called library to hold accession number, title of the book, author name, price of the book, and flag indicating whether book is issued or not.

- ✓ 13. A record contains name of cricketer, his age, and number of test matches that he has played and the average runs that he has scored in each test match. Create an array of structure to hold records of 20 such cricketer and then write a program to read these records and arrange them in ascending order by average runs.
- ✓ 14. WAP to 2D array with values 0 & 1. WAP to sort each row and then find the row with the maximum number of 1s.

Example:

Input matrix

1 1 1 0

1 0 1 0

1 1 1 1

0 0 0 0

Sorted matrix

0 1 1 1

0 0 1 1

1 1 1 1 // this row has maximum 1s

0 0 0 0

Output: 2

- ✓ 15. WAP to input a two dimensional array and print lower triangular matrix and upper triangular matrix. Lower triangular matrix is a matrix which contains elements below principle diagonal including principle diagonal elements and rest of the elements are 0. Upper triangular matrix is a matrix which contains elements above principle diagonal including principle diagonal elements and rest of the elements are 0.
- ✓ 16. WAP to input two matrices A and B. The task is to multiply matrix A and matrix B recursively. If matrix A and matrix B are not multiplicative compatible, then generate output "Not Possible".
- ✓ 17. WAP to input the value for 1 D array. Write a function leftRotate(ar[], d, n) that left rotates the array of size n by d elements.

Input array:

1	2	3	4	5	6	7
---	---	---	---	---	---	---

After Left Rotation:

3	4	5	6	7	1	2
---	---	---	---	---	---	---



- ✓ 18. WAP to input the value for 1 D array. Write a function rightRotate(ar[], d, n) that right rotates the array of size n by d elements.
- ✓ 19. WAP to sort an array in the wave form. An array "arr[0..n-1]" is sorted in wave form if $arr[0] \geq arr[1] \leq arr[2] \geq arr[3] \leq arr[4] \geq \dots$

Input: arr[] = {10, 5, 6, 3, 2, 20, 100, 80}
Output: arr[] = {10, 5, 6, 2, 20, 3, 100, 80}
OR
{20, 5, 10, 2, 80, 6, 100, 3} OR
any other array that is in wave form

- ✓ 20. WAP to input values for 1-D array that may contain both positive and negative integers, find the sum of contiguous sub-array of numbers which has the largest sum. For example, if the given array is {-2, -5, 6, -2, -3, 1, 5, -6}, then the maximum sub-array sum {-2, -5, **6**, **-2**, **-3**, **1**, **5**, -6} is 7 (see dark red bold elements).
- ✓ 21. WAP to print the following pattern wherein the input is the character.

A
AB
ABC
ABCD
ABCDE
ABCDEF

- ✓ 22. WAP to print the following pattern wherein the input is the number.

1
22
333
4444
55555

- ✓ 23. WAP to print the following pattern wherein the input is the number.

1
12
123
1234
12345
123456

- ✓ 24. WAP to enter a decimal number and calculate & display the binary equivalent of the number.

- ✓ 25. WAP to reverse a string using recursion.

26. WAP to concatenate three strings using recursion.

- ✓ 27. WAP to compute $F(m, n)$ where $F(m, n)$ can be recursively defined as

$F(m, n) = 1$ if $m=0$ or $m \geq n \geq 1$

$F(m, n) = F(m, n-1) + F(m-1, n-1)$ otherwise

- ✓ 28. WAP to merge two integer arrays and display the merged array. The merge array should be in arranged in descending order.

- ✓ 29. WAP to fill a square matrix with value 0 on the diagonal elements, 1 on the upper diagonal elements and -1 on the lower diagonal elements.



- ✓ 30. WAP to build an array of 100 random numbers in the range of 1 to 100. Perform the following operation in the array.
 - a. Count the number of elements that are completely divisible by 5
 - b. Count the number of odd elements
 - c. Find the smallest element in the array
 - d. Find the position of the largest value in the array
- 31. WAP to transpose $n \times n \times n$ matrix.
- 32. WAP to replace a pattern with another pattern in the text. E.g. If the input text is "Hi, I am doing good at home" and the replace pattern is "Home" and the another pattern is "hostel", the new text would be "Hi, I am doing good at hostel"
- ✓ 33. WAP to enter a text that has columns. Replace all the commas with semi colons and then display the text.
- 34. WAP to insert a word before a given word in the text.
- 35. WAP to enter points in three dimensional spaces and then calculate the distance between them.
- 36. WAP to read, add, subtract, multiply, divide, display two heights. Height is defined using feet and inches.
- ✓ 37. WAP to input the string expression. Check whether the pairs and the orders of "{", "}", "(", ")", "[", "]" are correct in expression. For example, the program should print true for exp = "[(){}{[(())()}]}" and false for exp = "[()]"
- 38. WAP to input a decimal number m. Convert it in binary string and apply n iterations, in each iteration 0 becomes 01 and 1 becomes 10. Find kth character in the string after nth iteration.
- 39. WAP to input an array of distinct integers and a sum value. Find count of any triplets with sum smaller than given sum value.
- 40. WAP to input two unsorted arrays of integer arr1[] and arr2[]. Each array may contain duplicates. For each element in arr1[] count elements less than or equal to it in array arr2[].
- 41. WAP to find the kth min element from a 2-D array.
- 42. WAP to input array containing integers, zero is considered an invalid number and rest all other numbers are valid. If two nearest valid numbers are equal, then double the value of the first one and make the second number as 0. At the end, move all the valid numbers on the left.
- ✓ 43. Given a string consisting of only 0, 1, A, B, C where
A = AND, B = OR, C = XOR
Calculate the value of the string assuming no order of precedence and evaluation is done from left to right.
- 44. WAP to input array of penalties, an array of car numbers and also the date. The task is to find the total fine which will be collected on the given date. Fine is collected from odd-numbered cars on even dates and vice versa.
- 45. WAP to input two integers n and m. The problem is to find the number closest to n and divisible by m. If there is more than one such number, then output the one having maximum absolute value. If n is completely divisible by m (not equal to 0), then output n only. Example –
 - a. If n = 13 and m = 4 then output is 12.
 - b. If n = 15 and m = 4 then output is 16.
 - c. If n = -15 and m = 6 then output is either -18 or -12.



46. WAP to input two strings. Modify 1st string such that all the common characters of the 2nd strings have to be removed and the uncommon characters of the 2nd string have to be concatenated with uncommon characters of the 1st string. If the modified string is empty then print '-1'.
47. WAP to input a string. The string is composed of lowercase alphabets and wildcard characters i.e. '?'. Here, '?' can be replaced by any of the lowercase alphabets. Now you have to classify the given string as good or bad on the basis of following rule - If there are more than 3 consonants together OR more than 5 vowels together, the string is considered to be "BAD". A string is considered "GOOD" only if it is not "BAD".
- ✓ 48. WAP to input two arrays of equal length, the task is to find if given arrays are equal or not. Two arrays are said to be equal if both of them contain same set of elements, arrangements (or permutation) of elements may be different though.
Example 1 - A[] = {1, 2, 5, 4, 0}; B[] = {2, 4, 5, 0, 1}; then Output : 1
Example 2 - A[] = {1, 2, 5}; B[] = {2, 4, 15}; then Output : 0
- ✓ 49. Ugly numbers are numbers whose only prime factors are 2, 3 or 5. The sequence 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, ... shows the first 11 ugly numbers. By convention, 1 is included. WAP to find nth ugly number. n to be input from the keyboard.
- ✓ 50. WAP that calculates the day of the week for any particular date in the past or future.

