PERFORM ALL QUESTIONS WHICH ARE POSSIBLE FOR A PROGRAMMING LANGUAGE TO DO.

|  |  |
| --- | --- |
| 1 | WAP TO CHECK IF A NUMBER IS ARMSTRONG OR NOT |
| 2 | WAP TO MAKE INFINITE LOOP FROM ALL TYPES OF LOOPS. |
| 3 | WAP TO FIND HIGHEST OF THE THREE ENTERED NUMBERS. |
| 4 | WAP TO DISPLAY THE SUM OF ITS DIGITS |
| 5 | WAP TO FIND FACTORIAL OF A NUMBER |
| 6 | WAP TO FIND FACTORIAL USING RECURSION |
| 7 | WAP TO FIND GCD OF A NUMBER. |
| 8 | WAP TO FIND GCD OF A NUMBER USING RECURSION |
| 9 | WAP TO PRINT THE FIBONACCI SERIES (BY USER INPUT) |
| 10 | WAP TO REVERSE A NUMBER |
| 11 | WAP TO CHECK IF THE ENTERED NUMBER IS EVEN, ODD OR PRIME |
| 12 | WAP TO CHECK IF THE ENTERED NUMBER IS PALINDROME OR NOT |
| 13 | WAP TO CHECK IF THE ENTERED STRING IS PALINDROME OR NOT |
| 14 | WAP TO MAKE A VARIABLE ARGUMENT FUNCTION. |
| 15 | WAP TO PRINT THE FIBONACCI SERIES USING RECURSION. |
| 16 | WAP TO SWAP TWO NUMBER USING CALL BY VALUE |
| 17 | WAP TO SWAP TWO NUMBERS USING CALL BY REFERENCE. |
| 18 | WAP TO SWAP TWO NUMBER WITHOUT TAKING 3RD VARIABLE. |
| 19 | WAP TO CONVERT INTEGER TO STRING AND VICE VERSA |
| 20 | WAP TO MAKE A RAPID CALCULATOR |
| 21 | WAP TO COUNT NUMBER OF VOWEL, CONSONANTS, NUMBER, SPACE, SPECIAL CHARACTER FROM A STRING INPUT. |
| 22 | WAP TO FIND LCM OF A NUMBER. |
| 23 | WAP TO REMOVE WHITESPACE FROM STRING |
| 24 | WAP TO IMPLEMENT ARRAY (INSERTION, DELETION, DISPLAY AND SEARCHING). |
| 25 | WAP TO FIND SUM OF ALL ELEMENTS IN ARRAY. |
| 26 | WAP TO COUNT EVEN, ODD AND PRIME NUMBER FROM AN ARRAY. |
| 27 | WAP TO FIND LARGEST AND SMALLEST ELEMENTS IN AN ARRAY. |
| 28 | WAP TO FIND SECOND LARGEST ELEMENT IN AN ARRAY |
| 29 | WAP TO SWAP AN ARRAY |
| 29 | WAP TO COUNT REPEATED ELEMENTS IN AN ARRAY |
| 30 | WAP TO CHECK IF THE ARRAY IS PALINDROME OR NOT |
| 31 | WAP TO ROTATE AN ARRAY (LEFT AND RIGHT). |
| 32 | WAP TO MERGE TWO ARRAYS |
| 33 | WAP TO PRINT A 2D ARRAY |
| 34 | WAP TO ADD TWO MATRICES |
| 35 | WAP TO PRINT DIAGONAL ELEMENTS OF MATRIX. |
| 36 | WAP TO PRINT SUM OF DIAGONAL ELEMENTS OF MATRIX |
| 37 | WAP TO PRINT TRANSPOSE OF MATRIX |
| 38 | WAP TO MULTIPLY TWO MATRIX |
| 39 | WAP TO MAKE A USER DEFINED JAGGED ARRAY AND DISPLAY IT USING FOREACH LOOP. |
| 40 | WAP TO PRINT ASCII VALUE OF ALL CHARACTERS. |
| 41. | WAP TO IMPLEMENT DYNAMIC ARRAY. |
| 42 | WAP TO IMPLEMENT STACK USING ARRAY |
| 43 | WAP TO IMPLEMENT QUEUE USING ARRAY. |
| 44 | WAP TO IMPLEMENT CIRCULAR QUEUE USING ARRAY |
| 45 | WAP TO MAKE A TABLE MAKER. |
| 46 | WRITE A FUNCTION TO POWER A VALUE WITHOUT USING IN-BUILT METHODS. |
| 47 | WAP TO FACTORISE A NUMBER GIVEN BY USER |
| 48 | WAP TO IMPLEMENT LINEAR LINKED LIST. |
| 49 | WAP TO IMPLEMENT CIRCULAR LINKED LIST |
| 50 | WAP TO IMPLEMENT DOUBLY LINKED LIST |
| 51 | WAP TO IMPLEMENT STACK USING LINKED LIST |
| 52 | WAP TO IMPLEMENT QUEUE USING LINKED LIST. |
| 53 | WAP TO DISPLAY A LINKED LIST IN REVERSE USING ARRAY |
| 54 | WAP TO DISPLAY A LINKED LIST IN REVERSE WITHOUT ARRAY. |
| 55 | WAP TO CONVERT A LINKED LIST INTO A STRING. |
| 56 | WAP TO REVERSE A LINKED LIST. |
| 57 | WAP TO IMPLEMENT FIND VALUE OF x from a quadratic equation. |
| 58 | WAP TO PLAY AN AUDIO. |
| 59 | WAP TO SORT A LINKED LIST. |
| 60 | WAP TO CONVERT BINARY TO DECIMAL AND VICE VERSA |
| 61 | WAP TO MAKE  \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \* |
| 62 | WAP TO MAKE  \* \* \* \* \*  \* \* \* \*  \* \* \*  \* \*  \* |
| 63 | WAP TO MAKE  \* \* \* \* \*  \* \* \* \*  \* \* \*  \* \*  \* |
| 64 | WAP TO MAKE  \* \* \* \* \* \* \* \* \* \* \*  \* \*  \* \*  \* \*  \* \* \* \* \* \* \* \* \* \* \* |
| 65 | WAP TO MAKE  1  0 1  0 1 0  1 0 1 0  1 0 1 0 1 |
| 66 | WAP TO MAKE  \* \*  \* \* \* \*  \* \* \* \* \* \*  \* \* \* \* \* \* \* \*  \* \* \* \* \* \* \* \*  \* \* \* \* \* \*  \* \* \* \*  \* \* |
| 67 | WAP TO MAKE  \*  \* \* \*  \* \* \* \* \*  \* \* \* \* \* \* \*  \* \* \* \* \* \* \*  \* \* \* \* \*  \* \* \*  \* |
| 68 | WAP TO MAKE  1  2 3  4 5 6  7 8 9 10  11 12 13 14 15 |
| 69 | WAP TO MAKE  1 2 3 4 5  1 2 3 4  1 2 3  1 2  1 |
| 70 | WAP TO MAKE  1  2 1 2  3 2 1 2 3  4 3 2 1 2 3 4  5 4 3 2 1 2 3 4 5 |
| 71 | WAP TO MAKE  1  2 2  3 3 3  4 4 4 4  5 5 5 5 5 |
| 72 | WAP TO MAKE  \* \* \* \* \*  \* \* \* \* \*  \* \* \* \* \*  \* \* \* \* \*  \* \* \* \* \* |
| 73 | WAP TO CHECK IF THE ENTERED NUMBER IS PERFECT OR NOT. |
| 74 | WAP TO CHECK IF THE ENTERD NUMBER IS HAPPY OR NOT. |
| 75 | WAP TO REPLACE ALL VOWELS IN THE ENTERED STRING AS “NULL”.  e.g. Aman - NULLmNULLn |
| 76 | WAP TO BUBBLE SORT AN ARRAY. |
| 77 | WAP TO INSERTION SORT AN ARRAY |
| 78 | WAP TO SELECTION SORT AN ARRAY |
| 79 | WAP TO MERGE SORT AN ARRAY |
| 80 | WAP TO QUICK SORT AN ARRAY |
| 81 | WAP TO HEAP SORT AN ARRAY |
| 82 | WAP TO IMPLEMENT VECTOR |
| 83 | WAP TO IMPLEMENT BINARY TREE (INSERTION, DELETION, IN-ORDER, PRE-ORDER, POST-ORDER AND SEARCHING). |
| 84 | WAP TO IMPLEMENT BINARY SEARCH TREE |
| 85 | WAP TO IMPLEMENT AVL TREE. |
| 86 | WAP TO IMPLEMENT B+ TREE. |
| 87 | WAP TO IMPLEMENT B- TREE. |
| 88 | WAP TO FIND THE HEIGHT OF THE TREE |
| 89 | WAP TO IMPLEMENT PAIR USING VECTOR |
| 90 | WAP TO IMPLEMENT LIST. |
| 92 | WAP TO IMPLEMENT MAPS |
| 93 | WAP TO IMPLEMENT SETS. |
| 94 | WAP TO IMPLEMENT MULTISET |
| 95 | WAP TO IMPLEMENT PRIORITY QUEUE |
| 96 | WAP TO IMPLEMENT UNORDERED SET |
| 97 | WAP TO IMPLEMENT HASH-SET |
| 98 | WAP TO IMPLEMENT HASH-MAP |
| 99 | WAP TO IMPLEMENT HASH-TABLE |
| 100 | WAP TO MAKE GENERIC CALCULATOR. |
| 101 | WAP TO IMPLEMENT BUBBLE SORT USING GENERIC PROGRAMMING |
| 102 | WAP TO MAKE  \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \* |
| 103 | WAP TO MAKE  \* \* \* \* \*  \* \* \* \*  \* \* \*  \* \*  \* |
| 104 | WAP TO TAKE NO OF DAYS AS INPUT AND CONVERT IT INTO YEAR/MONTH/DAY. |
| 105 | WAP TO GENERATE RANDOM INTEGER AND FLOAT VALUE. |
| 106 | WAP TO CALCULATE THE TIME TAKEN BY YOUR CODE TO EXECUTE. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |