**COMPUTER APPLICATION PROJECT- CLASS X**

**SESSION 2017-18**

**Execute the following programs and submit the documentation of the program latest by 10th Nov’17. Write Objective, Algorithm, Source code, Output and VDT for each program.**

**Q1.Display the series : 1+21+321+4321+..........+n terms(Preferably by arithmetic method).**

**Q2. Using Scanner class, write a program to accept a sentence and display only 'Palindrome' words.**

**eg. Sample Input : MADAM ARORA IS OUT ENGLISH TEACHER.**

**Sample Output : MADAM**

**ARORA**

**Q3. WAP to check whether a number is a Pronic Number or not .**

**Pronic Number : A pronic number, oblong number, rectangular number or heteromecic number, is a number which is the product of two consecutive integers, that is, n (n + 1).  
0, 2, 6, 12, 20, 30, 42, 56, 72, 90, 110, 132, 156, 182, 210, 240, 272, 306, 342, 380, 420, 462 … etc.**

**Q4Write a menu driven program to enter a number and check the following:**

1. **Armstrong number**
2. **Prime number**
3. **Disarium number**   
   (135 is a DISARIUM, 11+32+53 = 135, some other **DISARIUM** are 89, 175, 518 etc)

**Q5.  Write a program to check whether the number entered by the user is a Unique number or not. A Unique number is one that has no repetition of digits. i.e. all the digits are distinct and appears only once. For e.g. 1234, 65234 etc… whereas digits like 1000, 65536, 121 are not.**

**Q6. Input a number and check whether the number entered by the user is an Evil number or not. An Evil number is one whose Binary counterpart has even number of ones. E.g. 10, as the binary of 10 is 1010 which has 2 ones (hence Evil), whereas numbers that has Odd number of 1’s are called Odious numbers**

**Q7. WAP to create the following pattern:**

**\* \* \* \***

**\* \* \***

**\* \* \* \***

**\* \* \***

**\* \* \* \***

**\* \* \***

**\* \* \* \***

**Q8. Define a class named movieMagic with the following description:**

**Instance variables/data members:**

**int year            –           to store the year of release of a movie**

**String title       –           to store the title of the movie.**

**float rating      –           to store the popularity rating of the movie. (minimum rating = 0.0 and maximum rating = 5.0)**

**Member Methods:**

**(i)         movieMagic()              Default constructor to initialize numeric data members to 0 and String data member to “”.**

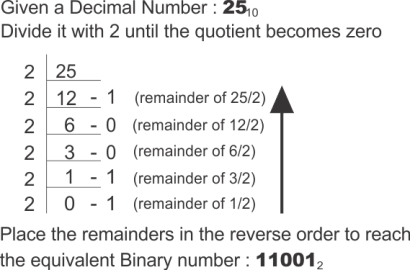
**(ii)        void accept()               To input and store year, title and rating.**

**(iii)       void display()              To display the title of a movie and a message based on the rating as per the table below.**

|  |  |
| --- | --- |
| **Rating** | **Message to be displayed** |
| **0.0 to 2.0** | **Flop** |
| **2.1 to 3.4** | **Semi-hit** |
| **3.5 to 4.5** | **Hit** |
| **4.6 to 5.0** | **Super Hit** |

**Q9. Write a Java program to accept a number and convert it into its binary form.**

**For Example: 25 in the Decimal number system can be represented as 11001 in the Binary number system.**

****

**Q10. To check whether the number is a magic number or not.  
[ e.g.  172 = 1+7+2 =10  
            10 = 1+0 = 1 ]  
( If the ultimate result is 1 then it is a magic number).**

**Q11. Write a program to input a word from the user and remove the duplicate characters present in it.**

**Example:**

**INPUT – abcabcabc  
OUTPUT – abc**

**INPUT – javaforschool  
OUTPUT – javforschl**

**Q12. Write a program that encodes a word into Piglatin. To translate a word into a Piglatin word, convert the word into uppercase and then place the first vowel of the original word as the start of the new word along with the remaining alphabets. The alphabets present before the vowel being shifted towards the end followed by “AY”.  
Sample Input (1) : London, Sample Output (1) : ONDONLAY  
Sample Input (2) : Olympics, Sample Output (2) : OLYMPICSAY**

**Q13. . Write a program to find the shortest and the longest word in a sentence and print them along with their length.**

**Sample Input: I am learning Java  
Sample Output:  
Shortest word = I  
Length = 1  
Longest word = learning  
Length = 8**

**Q14. Write a program in Java to accept a number and check whether it belongs to the Fibonacci Series (sequence) or not.**

**Fibonacci Series:  
The Fibonacci Sequence is the series of numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, …**

**The first two numbers in the series is ‘0’ and ‘1’ and every next number is found by adding up the two numbers before it.**

**The 2 is found by adding the two numbers before it (1+1)  
Similarly, the 3 is found by adding the two numbers before it (1+2),  
And the 5 is (2+3),  
and so on!**

**Example: the next number in the sequence above would be 21+34 = 55  
It is that simple!**

**Here is a longer list:**

**0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, …**

**Q15. Write a program in JAVA to find the Prime factors of a number.**

**Prime factors of a number are those factors which are prime in nature and by which the number itself is completely divisible (1 will not be taken as prime number).**

**Few such numbers are:  
Prime Factors of 24 are 2, 2, 2, 3  
Prime Factors of 6 are 2, 3**

**16. Mr.X has a savings bank account in BOI. He wants to make transaction from his account either by depositing some money or by making withdrawal.**

**Write a program to perform the above task with the following details.**

**Class name : *SavingsAccount***

**Data members :**

**String *name, ac\_type***

**int *acno***

**float *balance***

**Methods :**

***SavingsAccount(int bal)* : assign bal to balance**

***void deposit(int amt)* : deposits amt and update balance**

***void withdraw(int amt)* : withdraws amt and maintains balance. However, if**

**the balance becomes less than Rs. 500 then the**

**function should display an error message**

**“insufficient balance” and should not allow the**

**user to withdraw.**

***void display()* : displays balance of the account holder.**

**Q17. Define a class ‘*Telephone*’ having the following description:**

**Instance Variables / Data Members:**

**int *prv* - to store the previous and present meter reading int call**

**int *pre* - to store the calls made (i.e. pre – prv)**

**String *name* - to store name of the customer**

**double *amt* - to store the amount**

**double *total* - to store the total amount to be paid**

**Member Methods:**

***void input ( )*  - to input the previous reading, present reading and name of the customer**

***void cal ( )* - to calculate the amount and total amount to be paid**

***void display ( )* - to display the name of the customer, calls made, amount and total amount to be paid in the following format:**

**Name Calls Made Amount Total Amount**

**…….. ………….. ……….. ………………**

**Write a program to compute the monthly bill to be paid according to the given conditions:**

**Calls made Rate**

**Up to 100 calls No charge**

**For the next 100 calls 90 paise per call**

**For the next 200 calls 80 paise per call**

**More than 400 calls 70 paise per call**

**However every customer has to pay Rs. 180 per month as monthly rent for availing the service.**

## Q18. Write a Java program to print the following:

## Print Series 1  12    123    1234 …………n

## Print Series 1  11  111  111……..n terms

## Print Sum of Series 1/1^2  +  1/2^2  +  1/3^2 …………1/n^2

**Q19. An ISBN ( International Standard Book Number) is a ten digit code which uniquely identifies a book. The first nine digits represent the Group, Publisher and Title of the book and the last digit is used to check whether ISBN is correct or not. Each of the first nine digits of the code can take a value between 0 and 9. Sometimes it is necessary to make the last digit equal to ten; this is done by writing the last digit of the code as X. To verify an ISBN, calculate 10 times the first digit, plus 9 times the second digit, plus 8 times the third and so on until we add 1 time the last digit. If the final number leaves no remainder when divided by 11, the code is a valid ISBN.   
  
For example:   
  
1. 02011003311 = 10\*0 + 9\*2 + 8\*0 + 7\*1 + 6\*1 + 5\*0 + 4\*3 + 3\*3 + 2\*1 + 1\*1 = 55   
Since 55 leaves no remainder when divisible by 11, hence it is a valid ISBN.   
Design a program to accept a ten digit code from the user. For an invalid in out, display an   
appropriate message. Verify the code for its validity in the format specified below:   
Test your program with sample data and some random data.   
  
Example 1   
INPUT CODE : 0201530821   
OUTPUT : SUM = 99   
LEAVES NO REMAINDER - VALID ISBN CODE**

**Q20. Design a class to overload a function Joystring() as follows :**

**(i)void Joystring(String s, char ch1, char ch2) with one string and two character arguments that replaces the character argument ch1 with the character argument ch2 in the given string s and points the new string  
Example :  
Input value of s = “TECHNALAGY”  
ch1 = ‘A’ , ch2 = ‘O’  
Output : “TECHNOLOGY”**

**(ii)void Joystring(String s) with one string argument that prints the position of the first space and the last space of the given String s.  
Example :First Index : 5  
Last Index : 36**

**(iii)void Joystring(String s1, String s2) with two string arguments that combines the two strings with a space between them and prints the resultant string. Example :  
Input value of s1 = “COMMON WEALTH”  
s2 = “GAMES”  
Output : “COMMON WEALTH GAMES”**

**\*\*\*\*\*\***