Exercises for Computing Systems I 2014-2015

1. Calculate the maximum of an array of numbers.
2. Calculate the factorial of a number.
3. Sort an array of numbers.
4. Check if a given string is a palindrome or not. A palindrome is a word whose meaning may be interpreted the same way in either forward or reverse direction. Eg: Anna, rotator, madam, etc.
5. Enter two matrices and multiply them.
6. Write a program to detect if an entered character is a letter, number or a special character.
7. Write a program to accept a string of upper-case characters and convert in to lower-case characters, or the vice versa.
8. Write a program to perform various mathematical operations on a pair of numbers such as addition, subtraction, multiplication, division, etc. The program should ask the user to enter the operands (numbers to perform the operation on), as well as the operator (the type of operation to be performed). The functions that implement the operations must be implemented on a separate file, and the appropriate header file should be included in the main file.
9. Modify the previous program such that the operands are now passed to the program via command line.
10. Create a structure/class to represent each student of a course. There must be at least the following fields: name, NIE and marks. Create an array/list to store all the students and then calculate the average marks of the class; also the program should be able to sort students based on their marks into outstanding, remarkable, pass or fail.
11. Write a program which accepts a string from the user, checks if the accepted string is a number, and if so then converts the entered string to its integer equivalent.
12. Extend the previous program to be able to detect decimal numbers.
13. Difficult: Write a program that stores adjacent nodes of a network, with the distances between them, and to calculate the fastest route between two points. This can be made as part of the transportation network of the Community of Madrid.

14. Write a program to check if a given number is prime or not, and if not a prime then find all its factors.

15. Write a program that simulates cellular automate, in which a board of 10X10 elements (cells) are randomly initialized with ones and zeros. Each cell will maintain its value (zero or one) in the next generation, if more than two third of its neighbouring cells have a value equal to it, or flip its value (if zero change to one, else if one change to zero) if less that one third of its neighbouring cells have a value equal to it. If between one third and two third of its neighbouring cells have a value equal to it, then flip its value at a probability of 50%. Display the state of the board, in each generation, on the screen in the form of a matrix.

16. Easy: Write a program that accepts elements of a matrix of a predetermined size, and display it on the screen in matrix form.

17. Extend the previous program such that the size of the matrix is determined by the user during runtime.

18. Write a program to determines if a given phone number is that of a land line or of a mobile phone. If its a land line number, determine which province the number arrives from.

19. Easy: Write a program that accepts the dimensions of a parallelepiped, and then calculate its area and volume.

20. Write a program that accepts and stores name, first surname and last surname of a group of fixed number of people, then sort them alphabetically based on any of the three fields. Also the program should be able to delete names that begin with a certain letter in any of the three fields.

21. Difficult: Extend the above program to be able to accept names of variable number of people (dynamic memory) and the possibility of finding a string anywhere on any of the fields.

22. Difficult: Write a program that detects rhyming words in a given text.

23. Difficult: Extend the above program to now also count the number of syllables.

24. Write a program that can detect numbers in a given text, convert them to integer, do the necessary dynamic memory allocation and then save the converted integer in the allocated memory.

25. Fibonacci series is a series of numbers which follows the below mentioned pattern:

    0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

    Where each number is calculated by adding the previous two numbers.

- The first two numbers of this series will be 0 and 1.

- The fourth number (2) in the above series is calculated by adding the numbers in the second (1) and the third (1) position of the series.

- Similarly, the fifth number (3) is calculated by adding the numbers in the third (1) and the fourth (2) position of the series.

- And the sixth number (5) = (2+3).
- Etc.

  Write a program that creates a vector containing Fibonacci series numbers, and prints the same on the screen. To create this vector, write a function **fibonacci(x,y,v,n)**

  Where **x** and **y** would be the first two numbers of the series, and in-turn of the vector,

  v[0] = x v[1] = y

  Then, each subsequent element of the vector should be the sum of the previous two elements of the vector.

  v would be the vector that contains the series, and **n** would be the size of the series.

26. Extend the previous program by adding a new feature to sort the vector of Fibonacci series in descending order, and display the same on the screen.

27. Write a program to check if a number belongs to the Fibonacci series. To do so, you could use Binet's formula:

If **N** is a given number, then its part of the Fibonacci series if either **$5*N^2+4$** or **$5*N^2-4$** is a square of a natural number.

28. Easy: Write a function that accepts an upper and a lower limit, calculate the squares of each of the integer number between the upper and the lower limit [inclusive], and then print the sum of all the squares. Do so using recursion.

29. Medium: Accept an integer number along with its base [Binary, octal, decimal or hexadecimal], validate the inputs, then convert the number into a different base, as requested by the user, and then print the converted number.

30. Difficult: Write a function to accept an integer number[Up to five digits long], and display the number in words.

E.g.: (a) 65 – Sixty Five.

(b) 12345 – Twelve Thousand Three Hundred and Forty Five.

31. Difficult: Write a program to accept any date in the 21st century and output the day of the week this date falls on. Validate the input for number of days in a month, leap year, number of months in a year, etc.