MCA-I ('2019)
DEPARTMENT OF COMPUTER SCIENCE
ROLLWALA COMPUTER CENTER, GUJARAT UNIVERSITY
FUNDAMENTALS OF PROGRAMMING
ASSIGNMENT-III

---

1. Write a C program to accept two strings. Find a set of alphabets, which are common in both strings and print this set in alphabetical order. If no occurrence is found print an appropriate message. For e.g.

   Sample i/p: - Str1 = PREPARE FOR EXAM
   Str2 = GET GOOD MARKS

   Sample o/p: -
   Following are the alphabets which occur in both strings: A E M O R

2. Insert a word in a given string Given three numbers *S*, *P* and *k*. Your task is to find if there are integers n_1, n_2,...,n_k such that $n_1 + n_2 + ... + n_k = S$, $n_1 * n_2 * ... * n_k = P$. If such integers exist, print them out. If no such sequence of integers exist, then print "NO".

   For example if *S*=11, *P*=48 and *k*=3 then 3, 4 and 4 is a solution. On the other hand, if *S*=11, *P*=100 and *k*=3, there is no solution and you should print "NO".

   Sample input 1:
   ```
   11 48 3
   ```
   Sample output 1:
   ```
   3 4 4
   ```
   Sample input 2:
   ```
   11 100 3
   ```
   Sample output 2:
   ```
   NO
   ```

3. Write Do you like treasure hunts? In this problem you are to write a program to explore the following array for a treasure.

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

   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.

4. Write a C program that reads an unsorted list of names in an array without using in-build string functions. Sort the given array and then searches a given name in the array using binary search technique(for searching you can use in-built string functions.

5. Write a C program to generate the following "pyramid" of digits, using nested loops

```
                1
              232
            34543
          4567654
        567898765
      67890109876
    7890123210987
  890123454321098
90123456765432109
  890123454321098
    7890123210987
      67890109876
        567898765
          4567654
            34543
              232
                1
```

6. Write a C program that reads two arrays of strings, sort two arrays using selection sort technique. Now merge these two arrays into a single sorted array – removing duplicates if any.

7. In 1949 the Indian mathematician D.R. Kaprekar discovered a class of numbers called self-numbers. For any positive integer n, define d(n) to be n plus the sum of the digits of n. For example, d(75) = 75 + 7 + 5 = 87. Given any positive integer n as a starting point, you can construct the infinite increasing sequence of integers n, d(n), d(d(n)), d(d(d(n))), ... For example, if you start with 33, the next number is 33 + 3 + 3 = 39, the next is 39 + 3 + 9 = 51, the next is 51 + 5 + 1 = 57, and so you generate the sequence:

   33, 39, 51, 57, 69, 84, 96, 111, 114, 120, 123, 129, 141, ...

   The number n is called a generator of d(n). In the sequence above, 33 is a generator of 39, 39 is a generator of 51, 51 is a generator of 57, and so on. Some numbers have more than one generator: for example, 101 has two generators, 91 and 100. A number with no generators is a **self-number**. There are thirteen self-numbers less than 100: 1, 3, 5, 7, 9, 20, 31, 42, 53, 64, 75, 86, and 97.

   What is the total number of self numbers below N. N is to be read from user.

8. The standard Fibonacci sequence is represented by the sum of the previous 2 terms. Generally, it can be expressed as:

   $F_n = F_{n-1} + F_{n-2}$

   Just as well, we can express an X-nacci sequence where the next number is the sum of the previous X number of terms. The general form for this would be:

   $F_n = F_{n-1} + F_{n-2} + ... + F_{n-x}$

   The sequence for any X-nacci always begins with the first X terms being 1. So for a 4-Nacci sequence the start would be:

   1 1 1 1 4 7 13 25 ...

   Write a C program to generate M^{th} term of X-nacci series. Read M and X from the user.