**Q1. Write a C Program to find out the twin prime numbers between 1 to n?**

Input-->n=150. Output-->(3, 5), (5, 7), (11, 13), (17, 19), (29, 31), (41, 43), (59, 61), (71, 73), (101, 103), (107, 109), (137, 139)...

**Q2. Write a C program to Find Who should pay the bill among a group of friends?**

Note: Get number of friends and their Names at run time.

Consider 'n' friends going to a hotel and they are having a game among themselves on who should pay the bill.

The game is like this:

i. They should write their names and compare the letters present with all the names and remove the common letters in each names only once.

(Eg: RAM, JAYA, HARI, SITA – A is common in all but since it is present only one time in all names except JAYA (twice) only 1 A should be removed from JAYA)

ii. Then they should count the remaining number of letters available (RM + JYA + HRI + SIT = 11)

iii. Based on the count and using circular linked list eliminate one by one and find the last person standing who has to pay the bill. (Always start counting from the next person who got eliminated). Eg: RAM , JAYA, HARI, SITA

HARI is out

RAM is out

JAYA is out

SITA should pay the bill.

**Q3. Write a C program to solve below mentioned problem.**

A saint thought of teaching a mantra to one disciple. So he asks his disciple to collect some stones with different symbols on it.

The task is to arrange the stones in ascending order based on symbols on stones. Once the disciple is ready, saint will pick two symbols, the stones between those symbols should be arranged in descending order and remaining in ascending order. If he does that correctly he will be taught with mantra.

**Note:** Use single linked list to solve above problem Input-->1 9 2 8 3 7 4 6 5 symbol1-->2 symbol2-->7 Output-->1 2 6 5 4 3 7 8 9

**Q4. Write a function that takes integer input value and returns the immediate lowest number using the same digits?**

Ex1: Input: 321 Output: 312 Ex2: Input: 65438 Output: 65384

**Q5. In a microcontroller there are two 16 bit ports, namely port1 and port 2.**

**From $s^{th}$ position of port1 data copy n bits and write into $p^{th}$ position in port 2.**

**Write an API for the above problem.**

e.g: cpy_bits (unsigned int port1_address, int $s^{th}$_Position, int $p^{th}$_Position, int n_bits);