QUEUE

1. Queue | Set 2 (Linked List Implementation)
2. Circular Queue | Set 2 (Circular Linked List Implementation)
3. Implementation of Deque using doubly linked list.
4. Implementation of Deque using circular array
5. Implement Stack and Queue using Deque
6. Priority Queue using Linked List
7. Priority Queue using doubly linked list
8. reversing a Queue
9. Reversing a queue using recursion
10. Reversing the first K elements of a Queue
11. Interleave the first half of the queue with second half
12. Sorting a Queue without extra space
13. Implement Queue using Stacks
14. LRU Cache Implementation
15. Sliding Window Maximum (Maximum of all subarrays of size k)
16. Find the largest multiple of 3 | Set 1 (Using Queue)
17. Find the first circular tour that visits all petrol pumps
18. Smallest multiple of a given number made of digits 0 and 9 only
19. An Interesting Method to Generate Binary Numbers from 1 to n
20. Minimum time required to rot all oranges (Microsoft)
21. Sum of minimum and maximum elements of all subarrays of size k.
22. Distance of nearest cell having 1 in a binary matrix
23. First negative integer in every window of size k
24. Minimum sum of squares of character counts in a given string after removing k characters
25. Queue based approach for first non-repeating character in a stream (Microsoft)
26. Stack Permutations (Check if an array is stack permutation of other)
27. Check if X can give change to every person in the Queue

Solu. For Q. 17

```
int tour(petrolPump p[],int n)
{
   int s=0,sum=0,pre=0,i;

   for(i=0;i<n;i++)
   {
      sum+=(p[i].petrol-p[i].distance);
      if(sum<0)
      {
            pre+=sum;
            sum=0;
            s=i+1;
        }
    }
    return (sum+pre)>=0?s:-1;
}
```

priority_queue comparator

```
#include <iostream>
#include <queue>
using namespace std;

struct cmp
{
    bool operator()(const int &a,const int &b)
    {
        return (a<b);
    }
};


int main()
{
            priority_queue<int,vector<int>,cmp> q;
            for(int i=0;i<10;i++)
            q.push(i);

            while(!q.empty())
            {
                cout<<q.top()<<" ";
                q.pop();
            }
            return 0;
}
```