Day-7

1. https://leetcode.com/problems/count-and-say/
nahi bna

2. https://practice.geeksforgeeks.org/problems/longest-palindrome-in-a-string/0

```
#include<iostream>
#include<bits/stdc++.h>
using namespace std;
int main()
 {
        //code

        int t;
        cin>>t;
        while(t--)
        {
           string str;
           cin>>str;
           int len=str.size();
          int dp[len][len];
          memset(dp,0,sizeof(dp));
          for(int i=0;i<len;i++)
          {
             dp[i][i]=1;
          }

          int size=1,start=0;
          for(int i=0;i<len-1;i++)
          {
             if(str[i]==str[i+1])
             {
                if(size<2)
                {
                   size=2;
                   start=i;
                }
                dp[i][i+1]=1;
             }
          }


for(int i=3;i<=len;i++)
{
```

```
    for(int j=0;j<=len-i;j++)
    {
       int k=i+j-1;
       if(str[j]==str[k]&&dp[j+1][k-1])
       {dp[j][k]=1;
          if(size<i)
          {
             size=i;
             start=j;
          }
       }


    }

}
          for(int i=start;i<start+size;i++)cout<<str[i];
           cout<<endl;

          }
          return 0;
}
```

3.https://practice.geeksforgeeks.org/problems/longest-repeating-subsequence2004/1

```
          int LongestRepeatingSubsequence(string str){
                    // Code here
                    int len=str.size();
                    int dp[len+1][len+1];
                    for(int i=0;i<=len;i++)
                    {
                       dp[i][len]=0;
                       dp[len][i]=0;

                    }
                    for(int i=len-1;i>=0;i--)
                    {
                       for(int j=len-1;j>=0;j--)
                       {
                          if(str[i]==str[j]&&i!=j)dp[i][j]=dp[i+1][j+1]+1;
                          else
                          dp[i][j]=max(dp[i+1][j],dp[i][j+1]);
                       }
                    }
                    return dp[0][0];
                 }
```

4. https://www.geeksforgeeks.org/print-subsequences-string/

```
#include<iostream>
#include<bits/stdc++.h>
using namespace std;

void print(string input,string output)
{
   if(input.empty())
   {
      cout<<output<<" ";
      return;
   }
   print(input.substr(1),output+input[0]);//with include 1st char//
   print(input.substr(1),output);
}
int main()
{
  string output="";
  string input;
  cin>>input;
  print(input,output);
  return 0;
}
```

5. https://practice.geeksforgeeks.org/problems/permutations-of-a-given-string2041/1

Nahi bna

6. https://practice.geeksforgeeks.org/problems/median-in-a-row-wise-sorted-matrix1527/1#

```
class Solution{

public:

   int median(vector<vector<int>> &matrix, int r, int c){

      // code here

      vector<int>res;

      for(int i=0;i<matrix.size();i++)

      {

         for(int j=0;j<matrix[i].size();j++)

         {

            res.push_back(matrix[i][j]);

         }

      }

      sort(res.begin(),res.end());

      int len=res.size();

      if(len%2==1)
```

```
    {
        len=len/2+1;
    }
    else
    len/=2;
    return res[len-1];

  }
};
```

7. https://leetcode.com/problems/search-in-rotated-sorted-array/

```
class Solution {
public:
    int search(vector<int>& nums, int target) {
      if (nums.size() == 0) return -1;

      int left = 0, right = nums.size()-1;
       int start = 0;
       //1. find index of the smallest element
       while(left < right) {
          int mid = left + (right-left)/2;
          if (nums[mid] > nums[right]) {
             left = mid +1;
          } else right = mid;
       }

      //2. figure out in which side our target lies
       start = left;
       left = 0;
       right = nums.size()-1;
       if (target >= nums[start] && target <= nums[right])
          left = start;
       else right = start;

      //3. Run normal binary search in sorted half.
       while(left <= right) {
          int mid = left + (right - left)/2;
          if (nums[mid] == target) return mid;

          if (nums[mid] > target) right = mid-1;
          else left = mid + 1;
       }

       return -1;
```

```
    }
};
```