Recursion
(Assignment Solutions)

Question 1 :

```
int binSearch(int arr[], int si, int ei, int key) {
   if(si > ei) {
       return -1;
   }

   int mid = si + (ei - si)/2;
   if(arr[mid] == key) {
       return mid;
   } else if(arr[mid] > key) { //left half call
       return binSearch(arr, si, mid-1, key);
   } else { //right half call
       return binSearch(arr, mid+1, ei, key);
   }
}
```

Question 2 :

```
void allOccurences(int arr[], int key, int i, int n) {
   if(i == n) {
       return;
   }

   if(arr[i] == key) {
       cout << i << " ";
   }
   allOccurences(arr, key, i+1, n);
}
```

Question 3 :

```
int countSubstrs(string str, int i, int j, int n) {
      if (n == 1) {
```

shoryavishnoi13@gmail.com

```
            return 1;
        }
        if (n <= 0) {
            return 0;
        }

        int res = countSubstrs(str, i + 1, j, n - 1) +
                countSubstrs(str, i, j - 1, n - 1) -
                countSubstrs(str, i + 1, j - 1, n - 2);

        if (str[i] == str[j]) {
            res++;
        }
        return res;
    }


int main() {
   string str = "abcab";
   int n = str.size();
   cout << countSubstrs(str, 0, n-1, n) <<endl;
   return 0;
}
```

Question 4 :

The Solution for this particular question has also been discussed here in Java : https://www.youtube.com/watch?v=u-HgzgYe8KA

At timestamp : 00:05

```
void towerOfHanoi(int n, string src, string helper, string dest) {
   if(n == 1) {
      cout << "transfer disk " << n << " from " << src << " to " << dest << endl;
      return;
   }
    //transfer top n-1 from src to helper using dest as 'helper'
   towerOfHanoi(n-1, src, dest, helper);
   //transfer nth from src to dest
   cout << "transfer disk " << n << " from " << src << " to " << helper << endl;
   //transfer n-1 from helper to dest using src as 'helper'
```

shoryavishnoi13@gmail.com

```
    towerOfHanoi(n-1, helper, src, dest);
}

int main() {
    int n = 4;
    towerOfHanoi(4, "A", "B", "C");
    return 0;
}
```

Question 5 :

```
long long power(long long a,long long b){

    if(b==0) return 1;

    long long half_power= power(a,b/2);
    if(b%2 ==0)
    return half_power*half_power % MOD;

    else
    return half_power*half_power % MOD * (a % MOD) % MOD ;
}

int countGoodNumbers(long long n) {
    long long ed;
    long long od;
    if(n&1){
        od=n/2;
        ed=(n+1)/2;
    }
    else{
        od=n/2;
        ed=n/2;
    }
    return ((power(5,ed)%MOD)*(power(4,od)%MOD))%MOD;
}
```

shoryavishnoi13@gmail.com