Date: 22.1.2024

Session Topic: Recursion

Task 1

Question: Fibonacci Number

Solution:

```
class Solution {

    public int fib(int n) {

        // if(n==0)

        // return 0;

        // else if(n==1)

        // return 1;

        // int a=0,b=1;

        // int sum=0;

        // for(int i=2;i<=n;i++)

        // {

        //     int temp=a+b;

        //     if(i==n)

        //     {

        //         sum=temp;

        //     }

        //     System.out.println(temp);

        //     a=b;

        //     b=temp;

        // }

        // return sum;
```

```
        if(n==0)

        return 0;

        else if(n==1)

        return 1;

        return fib(n-1)+fib(n-2);

    }

}
```

Date: 22.1.2024

Session Topic: Recursion

Task 2

Question: Power of Two

Solution:

```
class Solution {

    public boolean isPowerOfTwo(int n) {

        int count=0;

        if(n==1 || n==8)

        return true;

        for(int i=1;i<=Math.sqrt(n);i++)

        {

            if(Math.pow(2,i)<=n)

            count++;

        }

        if(n==Math.pow(2,count))

        return true;

        return false;
```

```
    }

}
```

Date: 22.1.2024

Session Topic: Recursion

Task 3

Question: Power of Two

Solution:

```
class Solution {

    public double myPow(double x, int n) {

        return Math.pow(x,n);

    }

}
```

Date: 22.1.2024

Session Topic: Recursion

Task 5

Question: Kth-symbol in Grammar

Solution:

```
class Solution {

    public int kthGrammar(int n, int k) {

    //     if(n==1 && k==1)

    //     return 0;

    //     int a=n;

    //     String s=String.valueOf(0);

    //     StringBuilder sb = new StringBuilder();
```

```
//     sb.append(s);
//     Map<String,Integer>map=new TreeMap<>();
//     int index=1;int ind=2;
//     while(n>=2)
//     {
//         String str="";
//         for(int i=0;i<sb.length();i++)
//         {
//             if(sb.charAt(i)=='0')
//             str+="01";
//             else
//             str+="10";
//         }
//         if(index==1)
//         sb.delete(0,1);
//         index++;
//         map.put(str,ind++);
//         sb.delete(0,sb.length()+sb.length());
//         sb.append(str);
//         n--;
//     }
//     if(a!=1)
//     {
//         map.put("0",1);
//     }
//     int val=0;
```

```
    //     String st="";
    //     for(Map.Entry<String,Integer>i:map.entrySet())
    //     {
    //         if(i.getValue()==a)
    //         {
    //             st+=i.getKey();
    //             char ch[] = st.toCharArray();
    //             System.out.print(Arrays.toString(ch));
    //             return Integer.parseInt(String.valueOf(ch[k-1]));
    //         }
    //         st="";
    //     }

    //     return 0;

    if(n==1)
    return 0;
    if(k%2==0)
    return kthGrammar(n-1,k/2)==0?1:0;
    return kthGrammar(n-1,(k+1)/2)==0?0:1;
    }
}
```

Date: 22.1.2024

Session Topic: Recursion

Task 4

Question: Elimination Game

Solution:

```java
class Solution {

    public int lastRemaining(int n) {

        boolean leftToRight = true;

        int remaining = n;

        int step = 1;

        int head = 1;



        while (remaining > 1) {

            if (leftToRight || remaining % 2 == 1) {

                head = head + step;

            }

            remaining = remaining / 2;

            step *= 2;

            leftToRight = !leftToRight;

        }



        return head;

    }

}
```

Date: 22.1.2024

Session Topic: Recursion

Task 4

Question: Power of 4

Solution:

```
class Solution {

    public boolean isPowerOfFour(int n) {

        if(n==1)

        {

            return true;

        }

        for(int i=1;i<=Math.sqrt(n);i++)

        {

            if(Math.pow(4,i)==n)

            {

                return true;

            }

        }

        return false;

    }

}
```

Date: 22.1.2024

Session Topic: Recursion

Task 7

Question: Permutation Sequence

Solution:

```
class Solution {

    private void rec(int nums[], List<Integer>list,
List<List<Integer>>ans,boolean arr[])
```

```
    {
        if(list.size()==nums.length)
        {
            ans.add(new ArrayList<>(list));
            return;
        }
        for(int i=0;i<nums.length;i++)
        {
            if(!arr[i])
            {
                arr[i]=true;
                list.add(nums[i]);
                rec(nums,list,ans,arr);
                list.remove(list.size()-1);
                arr[i]=false;
            }
        }
    }
    public String getPermutation(int n, int k) {
        int arr[]=new int[n];
        for(int i=1;i<=n;i++)
        arr[i-1]=i;
        List<List<Integer>>ans=new ArrayList<>();
        List<Integer>list=new ArrayList<>();
        boolean flag[]=new boolean[n];
        rec(arr,list,ans,flag);
```

```
        String s="";
        int index=1;
        for(List<Integer>i:ans)
        {
            if(index==k)
            {
                for(int j:i)
                {
                    s+=j;
                }
                break;
            }
            index++;
        }
        return s;
    }
}
```