Date: 18.1.2024

Session Topic: Array Prefix

Task 1

Question: Find Pivot Index

Solution:

```
class Solution {

    public int pivotIndex(int[] nums) {int right=0,left=0;

        for(int i=0;i<nums.length;i++)

        {

            right+=nums[i];

        }

        for(int i=0;i<nums.length;i++)

        {

            right-=nums[i];

            if(right==left)

            {

                return i;

            }

            left+=nums[i];

        }

        return -1;

    }

}
```

Date: 18.1.2024

Session Topic: Array Prefix

Task 3

Question: Running Sum of 1d Array

Solution:

```
class Solution {

    public int[] runningSum(int[] nums) {

        int arr[]=new int[nums.length];

        arr[0]=nums[0];

        for(int i=0;i<nums.length-1;i++)

        {

            int sum=0;

            for(int j=i+1;j>=0;j--)

            {

                sum+=nums[j];

            }

            arr[i+1]=sum;

        }

        return arr;

    }

}
```

Date: 18.1.2024

Session Topic:

Task 4

Question: Find the Highest Altitude

Solution:

```
class Solution {

    public int largestAltitude(int[] gain) {

        int arr[]=new int[gain.length+1];

        arr[0]=0;

        int index=1;

        int max=arr[0];

        for(int i=0;i<gain.length;i++)

        {

            arr[index]=arr[i]+gain[i];

            if(arr[index]>max)

            {

                max=arr[index];

            }

            index++;

        }

        return max;

    }

}
```

Date: 18.1.2024

Session Topic:

Task 5

Question: Minimum Size Subarray Sum

Solution:

```
class Solution {
```

```
    public int minSubArrayLen(int target, int[] nums) {

        int left=0,sum=0,min=Integer.MAX_VALUE;

        for(int right=0;right<nums.length;right++)

        {

            sum+=nums[right];

            while(sum>=target)

            {

                sum-=nums[left];

                min=Math.min(min,right-left+1);

                left++;

            }

        }

        if(min==Integer.MAX_VALUE)

        {

            return 0;

        }

        return min;

    }

}
```

Date: 18.1.2024

Session Topic:

Task 6

Question: Subarray Sum Equals K

Solution:

```
class Solution {
```

```java
    public int subarraySum(int[] nums, int k) {

        int count=0;

        int arr[]=new int[nums.length+1];

        arr[0]=0;

        int index=1;

        for(int i=0;i<nums.length;i++)

        {

            arr[index++]=arr[i]+nums[i];

        }

        for(int i=0;i<arr.length;i++)

        {

            for(int j=i+1;j<arr.length;j++)

            {

                if(arr[j]-arr[i]==k)

                {

                    count++;

                }

            }

        }

        return count;

    }

}
```

Date: 18.1.2024

Session Topic:

Task 2

Question: Minimum Value to Get Positive Step by Step Sum

Solution:

```
class Solution {

    public int minStartValue(int[] nums) {

        int min=0;

        int sum=0;

        for(int i=0;i<nums.length;i++)

        {

            sum+=nums[i];

            min=Math.min(sum,min);

        }

        return 1-min;

    }

}
```

Date: 18.1.2024

Session Topic:

Task 7

Question: Apply Operations to Make All Array Elements Equal to Zero

Solution:

```
class Solution {

    public boolean checkArray(int[] nums, int k) {

        int curr=0,n=nums.length;

        for(int i=0;i<n;i++)

        {

            if(curr>nums[i])
```

```
                return false;

            nums[i]-=curr;

            curr+=nums[i];

            if(i>=k-1)

                curr-=nums[i-k+1];

        }

        return curr==0;

    }

}
```