SOUMYADEEP MITRA 12019002001069 CSE_SEC_A

AIM:Write a program in Double ended queue and priority queue

SRC:

Main.java | Reverse_k_element_queue.java | ArrayDequeDemo.java

```java
package codeForces;

import java.util.*;
public class ArrayDequeDemo {
    public static void main(String[] args)
    {
        // Initializing an deque
        Deque<String> dq
                = new ArrayDeque<String>();

        // add() method to insert
        dq.add("EN:12019002001069");
        dq.addFirst( e: "SOUMYADEEP MITRA");
        dq.addLast( e: "SECTION: A");

        System.out.println(dq);
    }
}
```

Run: ArrayDequeDemo

```
/Library/Java/JavaVirtualMachines/jdk-11.0.8.jdk/Contents/Home/bin/java
[SOUMYADEEP MITRA, EN:12019002001069, SECTION: A]

Process finished with exit code 0
```

Main.java | Reverse_k_element_queue.java | ArrayDequeDemo.java | PriorityQueueDemo.java

```java
package codeForces;

import java.util.*;
import java.io.*;

public class PriorityQueueDemo {

    public static void main(String args[])
    {
        PriorityQueue<String> pq = new PriorityQueue<>();

        pq.add("SOUMYADEEP");
        pq.add("1069");
        pq.add("2ND SEC A");

        System.out.println(pq);
    }
}
```

Run: PriorityQueueDemo

```
/Library/Java/JavaVirtualMachines/jdk-11.0.8.jdk/Contents/Home/bin/java
[1069, SOUMYADEEP, 2ND SEC A]

Process finished with exit code 0
```

SOUMYADEEP MITRA 12019002001069 CSE_SEC_A

AIM:Given an array of both positive and negative integers, the task is to compute sum of minimum and maximum elements of all sub-array of size k.

Examples:

Input : arr[] = {2, 5, -1, 7, -3, -1, -2}

K = 4

Output : 18

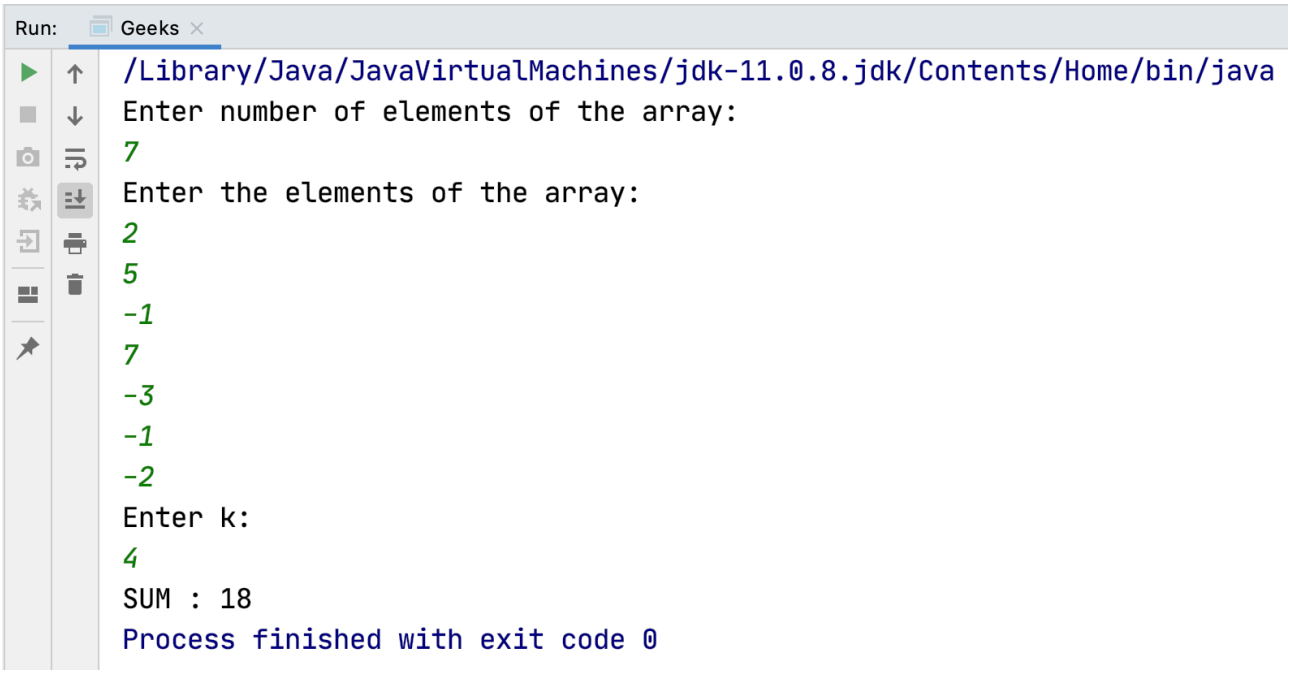
SRC:

Main.java | Reverse_k_element_queue.java | ArrayDequeDemo.java | PriorityQueueDemo.java | Geeks.java

```java
package codeForces;

import java.util.Deque;
import java.util.LinkedList;
import java.util.Scanner;

public class Geeks {

    public static int SumOfKsubArray(int arr[] , int k)
    {
        int sum = 0;

        Deque<Integer> S=new LinkedList<>(),G=new LinkedList<>();

        int i = 0;
        for (i = 0; i < k; i++)
        {
            while ( !S.isEmpty() && arr[S.peekLast()] >= arr[i])
                S.removeLast();

            while ( !G.isEmpty() && arr[G.peekLast()] <= arr[i])
                G.removeLast();


            G.addLast(i);
            S.addLast(i);
        }
```

Main.java | Reverse_k_element_queue.java | ArrayDequeDemo.java | PriorityQueueDemo.java | Geeks.java

```java

        for (  ; i < arr.length; i++ )
        {

            sum += arr[S.peekFirst()] + arr[G.peekFirst()];


            while ( !S.isEmpty() && S.peekFirst() <= i - k)
                S.removeFirst();
            while ( !G.isEmpty() && G.peekFirst() <= i - k)
                G.removeFirst();


            while ( !S.isEmpty() && arr[S.peekLast()] >= arr[i])
                S.removeLast();


            while ( !G.isEmpty() && arr[G.peekLast()] <= arr[i])
                G.removeLast();


            G.addLast(i);
            S.addLast(i);
        }

        sum += arr[S.peekFirst()] + arr[G.peekFirst()];

        return sum;
    }

    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of elements of the array: ");
        int ele = sc.nextInt();
        int[] arr = new int[ele];
        System.out.println("Enter the elements of the array: ");
        for(int h=0; h<ele; h++)
        {
            arr[h]=sc.nextInt();
        }
        System.out.println("Enter k: ");
        int k = sc.nextInt();
        System.out.print("SUM : ");
        System.out.print(SumOfKsubArray(arr, k));
    }
}
```


Run:
Geeks
/Library/Java/JavaVirtualMachines/jdk-11.0.8.jdk/Contents/Home/bin/java
Enter number of elements of the array:
7
Enter the elements of the array:
2
5
-1
7
-3
-1
-2
Enter k:
4
SUM : 18
Process finished with exit code 0