

```

/**Tamir Krief, Iaian Milton, Blessing Abumere */

public class Assignment2Task2 {

    public static void main(String[] args){
        java.util.Scanner input = new java.util.Scanner(System.in);

        int[] user_sequence = Prompt_Sequence(input);

        System.out.print("\nMax Decreasing Subsequence: " + MaxDecreasingSubsequence(user_sequence));
    }

    /**reads the initial sequence which is entered by the user*/
    public static int[] Prompt_Sequence(java.util.Scanner input){
        int[] sequence;
        try {
            System.out.println("\nEnter the length of the sequence");
            sequence = new int[input.nextInt()];

            System.out.println("\nEnter the values of the sequence[]");

            for (int i = 0; i < sequence.length; i++){
                sequence[i] = input.nextInt();
            }
            System.out.println("sequence[] = " + java.util.Arrays.toString(sequence));
            return sequence;
        }catch(java.util.InputMismatchException e){
            input.nextLine();
            System.err.print("Integer Values only. Please Try again\n");
            Prompt_Sequence(input);
        }

        return null;
    }

    /**exists to display results without the unit tests and it helped with the table results */
    public static void DisplayResults(){
        int[] seq1 = {4, 9,2,5,3,101,7,18,2,1};
        int[] seq2 = {186, 359, 274, 927, 890, 520, 571, 310, 916, 798, 732, 23, 196, 579,
            426,188, 524, 991, 91, 150, 117, 565, 993, 615, 48, 811, 594, 303, 191,
            505, 724, 818, 536, 416, 179, 485 , 334 , 74, 998, 100, 197, 768, 421,
            114, 739, 636, 356, 908 , 477, 65};
        int[] seq3 = {318 , 536 , 390 , 598 , 602 , 408 , 254 , 868 , 379 , 565 , 206 , 619 , 936 ,
195 ,
            123 , 314 , 729 , 608 , 148 , 540, 256 , 768 , 404 , 190 , 559 , 1000 , 482 , 141 , 26,
            230 , 550 , 881 , 759 , 122 , 878, 350, 756, 82, 562, 897, 508, 853, 317 ,
            380 , 807 , 23 , 506 , 98 , 757 , 247};

        System.out.print("\nMax Decreasing Subsequence:");

        System.out.printf(
            "\nSequence1: " + MaxDecreasingSubsequence(seq1) +
            "\nSequence2: " + MaxDecreasingSubsequence(seq2) +
            "\nSequence3: " + MaxDecreasingSubsequence(seq3)
        );
    }

    //same as Task1 but reversed?
    /** a program that computes the length of a longest decreasing subsequence of a sequence of
    integers. */
    public static int MaxDecreasingSubsequence(int[] arr){

```

```
//basecase
if(arr == null || arr.length == 0)
    return 0;

//d[i] corresponds with arr[i]
int[] d = new int[arr.length]; //keeps track of the COUNTS of the max decreasing subsequence

d[0] = 1; // d[0] will always be initialized to 1 as will any d[i] as that is the
basecase

int max_count = 1; //intitilization step is 1
for (int i = 1; i < arr.length;i++){ //i is always bigger than j
    d[i] = 1; //intitilized to 1 for the base case of no decreasing subsequences
    for (int j = 0; j < i; j++){
        //if current element is less than a previous element in the array and
        //if d[i] is less than d[j] + 1 then d[i] count will go up
        if (arr[i] < arr[j]){
            if (d[i] < d[j] + 1)
                d[i] = d[j] + 1;
        }
    }
    //update max_count if d[i] is bigger
    if (d[i] > max_count)
        max_count = d[i];
}

return max_count;

}

}
```