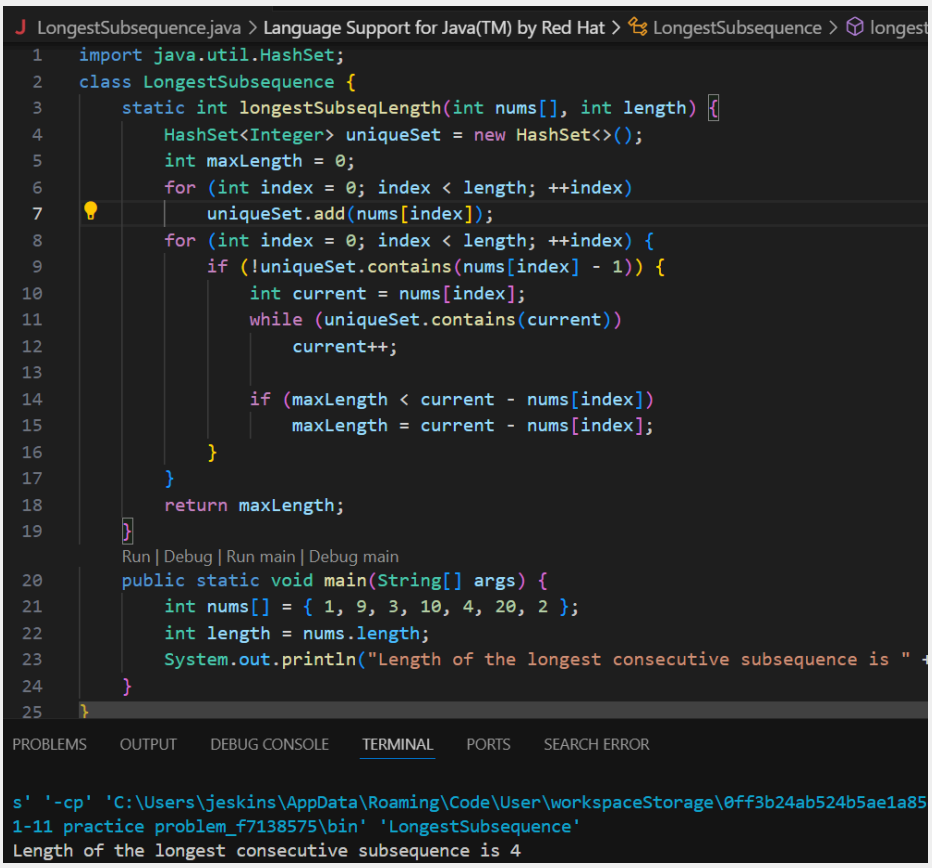
0-1 knapsack problem



```
LongestSubsequence.java > Language Support for Java(TM) by Red Hat > LongestSubsequence > longest
import java.util.HashSet;
class LongestSubsequence {
    static int longestSubseqLength(int nums[], int length) {
        HashSet<Integer> uniqueSet = new HashSet<>();
        int maxLength = 0;
        for (int index = 0; index < length; ++index)
            uniqueSet.add(nums[index]);
        for (int index = 0; index < length; ++index) {
            if (!uniqueSet.contains(nums[index] - 1)) {
                int current = nums[index];
                while (uniqueSet.contains(current))
                    current++;

                if (maxLength < current - nums[index])
                    maxLength = current - nums[index];
            }
        }
        return maxLength;
    }
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        int nums[] = { 1, 9, 3, 10, 4, 20, 2 };
        int length = nums.length;
        System.out.println("Length of the longest consecutive subsequence is " +
    }
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

s' '-cp' 'C:\Users\jeskins\AppData\Roaming\Code\User\workspaceStorage\0ff3b24ab524b5ae1a85
1-11 practice problem_f7138575\bin' 'LongestSubsequence'
Length of the longest consecutive subsequence is 4
```

Floor in Sorted Array

```
FloorInSortedArray.java > Java > FloorInSortedArray
import java.lang.*;
class FloorInSortedArray {
    static int floorSearch(int arr[], int n, int x)
    {
        if (x >= arr[n - 1])
            return n - 1;
        if (x < arr[0])
            return -1;
        for (int i = 1; i < n; i++)
            if (arr[i] > x)
                return (i - 1);
        return -1;
    }
    Run | Debug | Run main | Debug main
    public static void main(String[] args)
    {
        int arr[] = { 1, 2, 4, 6, 10, 12, 14 };
        int n = arr.length;
        int x = 7;
        int index = floorSearch(arr, n - 1, x);
        if (index == -1)
            System.out.print("Floor of " + x
                             + " doesn't exist in array ");
        else
            System.out.print("Floor of " + x + " is "
                             + arr[index]);
    }
}
```

```
PROBLEMS 2  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

s' '-cp' 'C:\Users\jeskins\AppData\Roaming\Code\User\workspaceStorage\0ff3b24ab52
1-11 practice problem_f7138575\bin' 'FloorInSortedArray'
Floor of 7 is 6
```

Check Equal Array

FloorInSortedArray.java 2 | EqualArray.java 4

EqualArray.java > Language Support for Java(TM) by Red Hat > EqualArray > main(String[])

```java
import java.io.*;
import java.util.*;
class EqualArray {
    public static boolean areEqual(int arr1[], int arr2[])
    {
        int N = arr1.length;
        int M = arr2.length;
        if (N != M)
            return false;
        Map<Integer, Integer> map
            = new HashMap<Integer, Integer>();
        int count = 0;
        for (int i = 0; i < N; i++) {
            if (map.get(arr1[i]) == null)
                map.put(arr1[i], value:1);
            else {
                count = map.get(arr1[i]);
                count++;
                map.put(arr1[i], count);
            }
        }
        for (int i = 0; i < N; i++) {
            if (!map.containsKey(arr2[i]))
                return false;
            if (map.get(arr2[i]) == 0)
                return false;
            count = map.get(arr2[i]);
            --count;
            map.put(arr2[i], count);
        }
        return true;
    }
```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

```
torage\0ff3b24ab524b5ae1a85dac9d8645f77\redhat.java\jdt_ws\11-11 practice proble
Yes equal array
```

PalindromeLinkedList

PalindromeLinkedList.java > Language Support for Java(TM) by Red Hat > Node >

```java
import java.util.Stack;
class Node {
    int data;
    Node next;
    Node(int d) {
        data = d;
        next = null;
    }}
class PalindromeLinkedList {
    static boolean isPalindrome(Node head) {
        Node currNode = head;
        Stack<Integer> s = new Stack<>();

        while (currNode != null) {
            s.push(currNode.data);
            currNode = currNode.next;
        }

        while (head != null) {
            int c = s.pop();
            if (head.data != c) {
                return false;
            }
            head = head.next;
        }

        return true;
    }
```

```
29      public static void main(String[] args) {
30          Node head = new Node(d:1);
31          head.next = new Node(d:2);
32          head.next.next = new Node(d:3);
33          head.next.next.next = new Node(d:2);
34          head.next.next.next.next = new Node(d:1);
35
36          boolean result = isPalindrome(head);
37
38          if (result)
39              System.out.println(x:"true");
40          else
41              System.out.println(x:"false");
42      }
43  }
44
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH

```
1-11 practice problem_f7138575\bin' 'PalindromeLinkedList'
true
```

Triplet Sum

```java
import java.util.Arrays;
import java.util.Scanner;
public class TripletArraySum {
    static boolean find3Numbers(int[] arr, int sum) {
        int n = arr.length;
        Arrays.sort(arr);
        for (int i = 0; i < n - 2; i++) {
            int l = i + 1;
            int r = n - 1;

            while (l < r) {
                int curr_sum = arr[i] + arr[l] + arr[r];
                if (curr_sum == sum) {
                    System.out.println(
                        "Triplet is " + arr[i] + ", "
                        + arr[l] + ", " + arr[r]);
                    return true;
                } else if (curr_sum < sum) {
                    l++;
                } else {
                    r--;
                }
            }
        }
        return false;
    }
```

```java
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s:"Enter the number of elements in the array: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];
        System.out.println(x:"Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }
        System.out.print(s:"Enter the desired sum: ");
        int sum = scanner.nextInt();
        if (!find3Numbers(arr, sum)) {
            System.out.println(x:"No triplet found");
        }
    }
}
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

```
Enter the number of elements in the array: 6
Enter the elements of the array:
12 3 4 1 6 9
Enter the desired sum: 24
Triplet is 3, 9, 12
```