HackerRank | Prepare Certify Compete Apply Search

All Contests > SJIT_Dream > Linked List - Adding a Task at a Specific Position

Linked List - Adding a Task at a Specific Position

Problem Submissions Leaderboard Discussions

You are developing a task tracker for a project where tasks are assigned unique IDs. These tasks are arranged in a linked list, based on the order in which they need to be completed. Sometimes, a new task needs to be inserted at a specific position in the list, ensuring it is prioritized correctly. Your task is to write a program to insert a new task at a given position in the task list and print the updated list after the insertion.

Write a program that inserts a new task (represented by a unique task ID) at a specific position in the task list. After the insertion, print the updated list of task IDs. If the entered position value is out the range, print the message as “Position out of range”.

Input Format

The first line contains an integer N, the number of task IDs initially added to the linked list. The second line contains N space-separated integers representing the task IDs. The third line contains an integer P, the position at which the new task ID should be inserted (1-based index). The fourth line contains an integer M, the new task ID to be inserted.

Constraints

The value of P will always be between 1 and N+1 (both inclusive).

Output Format

Print the updated task list with the elements separated by a space.

Sample Input 0

```
101 102 103 104
2
999
```

Sample Output 0

```
101 999 102 103 104
```

Explanation 0

The task list is initialized with 4 task IDs: 101 102 103 104. The new task ID 999 is inserted at position 2 (between 101 and 102). The updated list after insertion is 101 999 102 103 104.

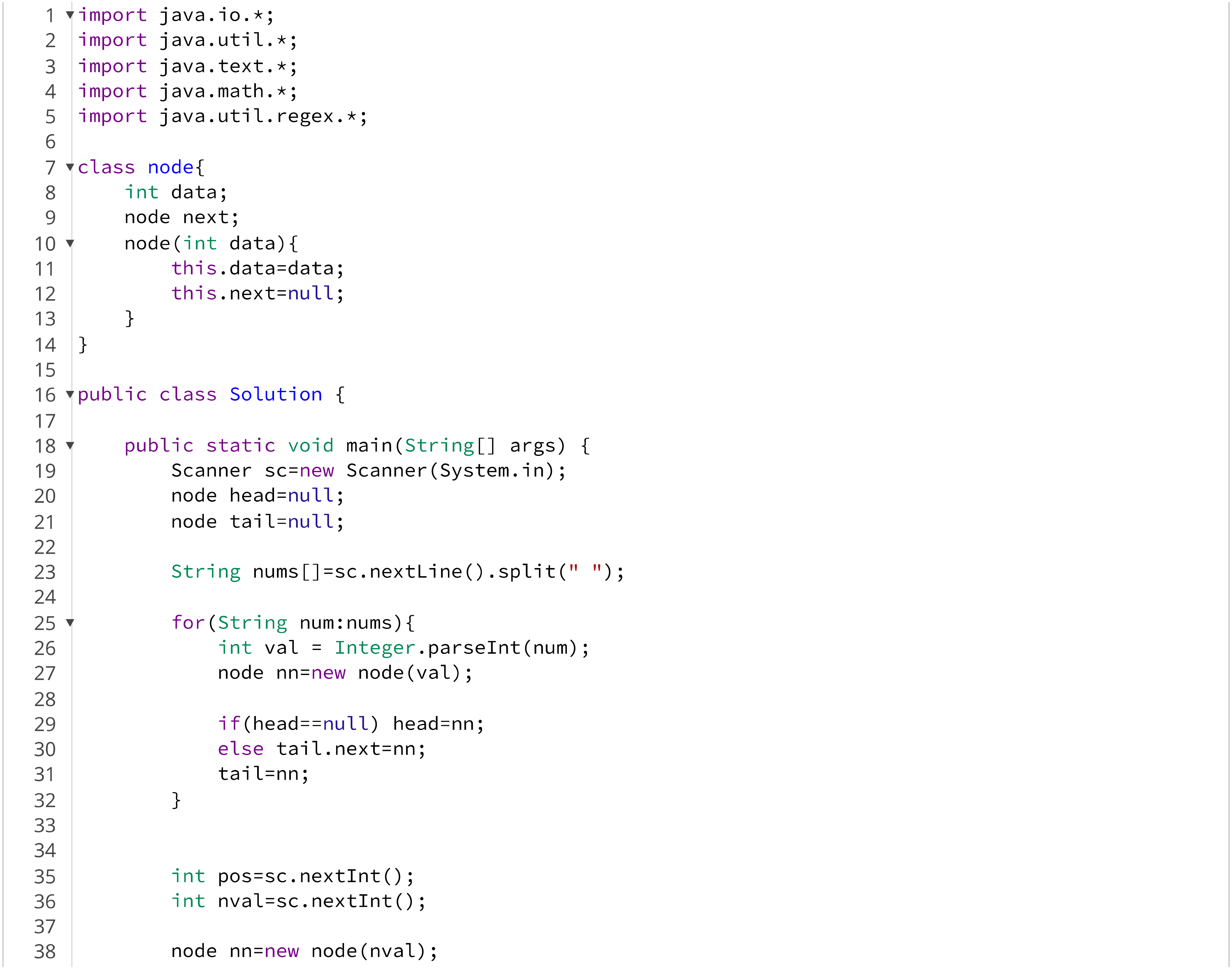
Contest ends in a month

Submissions: 35
Max Score: 10
Difficulty: Medium

Rate This Challenge:
☆☆☆☆☆

More

Java 7



```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

class node{
    int data;
    node next;
    node(int data){
        this.data=data;
        this.next=null;
    }
}

public class Solution {

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        node head=null;
        node tail=null;

        String nums[]=sc.nextLine().split(" ");

        for(String num:nums){
            int val = Integer.parseInt(num);
            node nn=new node(val);

            if(head==null) head=nn;
            else tail.next=nn;
            tail=nn;
        }


        int pos=sc.nextInt();
        int nval=sc.nextInt();

        node nn=new node(nval);
        node temp=head;
        if(pos==1){
            nn.next=head;
            head=nn;
        }
        else{
            for(int i=1;i<pos-1;i++){
                temp=temp.next;
            }
            nn.next=temp.next;
            temp.next=nn;
        }



        temp=head;
        while(temp.next!=null) {
            System.out.print(temp.data+" ");
            temp=temp.next;}
        System.out.print(temp.data);
    }
}
```

Line: 1 Col: 1

Upload Code as File

□ Test against custom input

Run Code

Submit Code