**Java Collections | Set 6 (Stack)**

Submissions: [594](https://practice.geeksforgeeks.org/problem_submissions.php?pid=3355)  Accuracy:

45.6%

   Difficulty: [Easy](https://practice.geeksforgeeks.org/Easy/0/0/)   Marks: 2

Show Topic Tags   

Java provides an inbuilt object type called **Stack**. It is a collection that is based on the last in first out (LIFO) principle. On Creation, a stack is empty. Try this problem using Stack.

Given **n** elements of a stack , delete the middle element of the stack without using any additional data structure.

**Input:** First line of input contains a single integer **T**which denotes the number of test cases. **T** test cases follows, first line of each test case contains a integer **n**. Second line consists of **n** spaced integers.

**Output:** Print the elements of the stack after deleting the middle element.

**Constraints:**

1<=T<=100  
1<= n <=103  
1<=Element<=104

**Example:**

**Input:**  
2  
5  
1 2 3 4 5  
6   
1 4 9 2 6 5

**Output:**  
5 4 2 1  
5 6 2 4 1

\*\* For More Input/Output Examples Use ['Expected Output'](https://practice.geeksforgeeks.org/problems/java-collections-set-1-stack/0/?ref=self#ExpectOP) option \*\*

Contributor: S\_A  
[Author: saksham0751](https://auth.geeksforgeeks.org/user/saksham0751/practice/)

<https://practice.geeksforgeeks.org/problems/java-collections-set-1-stack/0/?ref=self>

*/\**

*\* To change this license header, choose License Headers in Project Properties.*

*\* To change this template file, choose Tools | Templates*

*\* and open the template in the editor.*

*\*/*

**package** javaapplication43;

**import** java.util.\*;

**import** java.lang.\*;

**import** java.io.\*;

**public** **class** JavaApplication43 {

**static** **class** FastReader

    {

        BufferedReader br;

        StringTokenizer st;

**public** FastReader()

        {

            br = **new** BufferedReader(**new**

                     InputStreamReader(System.in));

        }

        String next()

        {

**while** (st == **null** || !st.hasMoreElements())

            {

**try**

                {

                    st = **new** StringTokenizer(br.readLine());

                }

**catch** (IOException  e)

                {

                    e.printStackTrace();

                }

            }

**return** st.nextToken();

        }

**int** nextInt()

        {

**return** Integer.parseInt(next());

        }

**long** nextLong()

        {

**return** Long.parseLong(next());

        }

**double** nextDouble()

        {

**return** Double.parseDouble(next());

        }

        String nextLine()

        {

            String str = "";

**try**

            {

                str = br.readLine();

            }

**catch** (IOException e)

            {

                e.printStackTrace();

            }

**return** str;

        }

    }

**public** **static** **void** main(String[] args) {

*// TODO code application logic here*

*//int[] arr = { 1 ,2, 3, 4, 5 };*

        FastReader fr = **new** FastReader();

**int** t = fr.nextInt();

**while**(t-- > 0) {

            Stack<Integer> s = **new** Stack();

**int** n = fr.nextInt();

            String[] input = fr.nextLine().trim().split(" ");

**int**[] arr = **new** **int**[n];

**for**(**int** i =0; i<n; i++) {

                arr[i] = Integer.parseInt(input[i]);

            }

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

                s.add(arr[i]);

            }

**if**(s.size()%2 != 0) {

                s.remove(n/2);

            }**else** {

                s.remove(n/2-1);

            }

*/\**

*for(int item : s){*

*System.out.print(item + " ");*

*}\*/*

**while**(!s.empty()) {

                System.out.print(s.pop() + " ");

            }

            System.out.println();

        }

    }

}