

<https://course.acciojob.com/idle?question=c44249e8-0776-4bcf-883d-fbdad0c956a9>

● EASY

● Max Score: 30 Points

- 
- 
- 
- 

## Insert node in a doubly linked list

You are given a Doubly linked list and an integer  $\kappa$  where every node in the linked list contains two pointers `next` and `prev` which point to the next node and previous node in the list respectively. Your task is to insert the integer  $\kappa$  at the head of the given linked list.

You have to complete `insertAtHead` function which contains `head` node and integer `K` i.e. value to be added as inputs and you have to return new linkedlist formed after adding node

### Input Format

The first line contains an integer  $N$  and  $\kappa$ .

The next line contains  $N$  integers, the elements of the Doubly linked list.

### Output Format

Print the updated Doubly linked list in new line.

### Example 1

Input

```
5 2
1 2 3 4 5
```

Output

2 1 2 3 4 5

Explanation

The initial linked list is 1->2->3->4->5. Insert 2 at the position 0 i.e. head, which currently 1 has in it.  
The updated linked list is 2->1->2->3->4->5

## Example 2

Input

6 8  
7 6 1 5 4 9

Output

8 7 6 1 5 4 9

Explanation

The initial linked list is 7->6->1->5->4->9. Insert 8 at the position 0 i.e. head, which currently 7 has in it. The updated linked list is 8->7->6->1->5->4->9

## Constraints

1 <= N <= 1000

1 <= K <= 1000

### Topic Tags

- **Linked lists**

# My code

```
// n java
import java.util.*;
import java.lang.*;
import java.io.*;

class Node
{
    int data;
    Node next;
    Node prev;

    Node(int data)//, Node next, Node prev
    {
        this.data = data;
        this.next = next;
        this.prev = prev;
    }

    Node() {}
}

public class Main
{
    static void display(Node h)
    {
        Node p=h;
        while(p!=null)
        {
```

```

        System.out.print(p.data+" ");
        p=p.next;
    }
}

```

```

static Node push(int new_data,Node head)//add 1st position
{
    Node new_Node = new Node(new_data);
    new_Node.next = head;
    new_Node.prev = null;
    if (head != null)
        head.prev = new_Node;
    head = new_Node;
    return head;
}

```

```

    public static void main (String[] args) throws
java.lang.Exception
    {
        //your code here
        Scanner s=new Scanner(System.in);
        int n=s.nextInt();
        int k=s.nextInt();
        int arr[]=new int[n];
        for(int i=0;i<n;i++)
            arr[i]=s.nextInt();
        Node a=null;
        for(int i=0;i<n;i++)
            a=push(arr[n-1-i],a);
        //display(a);
    }
}

```

```
// System.out.println("");  
a=push(k,a);  
display(a);  
}  
}
```