

```
1  import java.util.Scanner;
2  class Calculator
3  {
4      int add(int... val)
5      {
6          int sum=0;
7          for(int a: val)
8          {
9              sum=sum+a;
10         }
11         return sum;
12     }
13     int sub(int... val)
14     {
15         int diff=0;
16         for(int a: val)
17         {
18             diff=diff-a;
19             if(a==val[0])
20             {
21                 diff=-diff;
22             }
23         }
24         return diff;
25     }
26     int mul(int... val)
27     {
28         int mul=1;
29         for(int a: val)
30         {
31             mul=mul*a;
32         }
33         return mul;
34     }
35     double div(int... val)
36     {
37         double div=1;
38         for(int a: val)
39         {
40             div=div/a;
41         }
42         return div;
43     }
44 }
45 public class Main
46 {
```

```

48     {
49         Calculator c=new Calculator();
50         Scanner sc=new Scanner(System.in);
51         System.out.println("Enter the size of arrays :");
52         int size=sc.nextInt();
53         int arr1[]=new int[size];
54         int arr2[]=new int[size];
55         char operator[]={'+', '-', '*', '/'};
56         System.out.println("Enter first array elements");
57         for(int i=0;i<size;i++)
58         {
59             arr1[i]=sc.nextInt();
60         }
61         System.out.println("Enter second array elements");
62         for(int i=0;i<size;i++)
63         {
64             arr2[i]=sc.nextInt();
65         }
66         System.out.println("Choose your operator to perform operation");
67         for(int i=0;i<operator.length;i++)
68         {
69             System.out.println(operator[i]);
70         }
71         char op=sc.next().charAt(0);
72         int flag=0;
73         for(int i=0;i<operator.length;i++)
74         {
75             if(op==operator[i])
76             {
77                 switch(i)
78                 {
79                     case 0 :
80                         System.out.println("Adding two arrays");
81                         for(int j=0;j<arr1.length;j++)
82                         {
83                             flag=1;
84                             System.out.println(arr1[j]+arr2[j]);
85                         }
86                         break;
87                     case 1 :
88                         System.out.println("Subtracting two arrays");
89                         for(int j=0;j<arr1.length;j++)
90                         {
91                             flag=1;
92                             System.out.println(arr1[j]-arr2[j]);
93                         }
94                         break;

```



```

80         System.out.println("Adding");
81         for(int j=0;j<arr1.length;
82         {
83             flag=1;
84             System.out.println(arr1[j]+arr2[j]);
85         }
86         break;
87     case 1 :
88         System.out.println("Subtraction");
89         for(int j=0;j<arr1.length;
90         {
91             flag=1;
92             System.out.println(arr1[j]-arr2[j]);
93         }
94         break;
95     case 2 :
96         System.out.println("multiplication");
97         for(int j=0;j<arr1.length;
98         {
99             flag=1;
100            System.out.println(arr1[j]*arr2[j]);
101        }
102        break;
103    case 3 :
104        System.out.println("Division");
105        for(int j=0;j<arr1.length;
106        {
107            flag=1;
108            System.out.println(arr1[j]/arr2[j]);
109        }
110        break;
111    }
112 }
113 }
114 if(flag==0)
115 {
116     System.out.println("Invalid operator");
117 }
118 }
119 }

```



Enter the size of arrays :

2

Enter first array elements:

37

36

Enter second array elements:

73

47

Choose your operator to calculate from the opti

+

-

*

/

+

Adding elements in both the arrays :

37+73=110

36+47=83

Process finished.

```
1
2 import java.util.*;
3 class Main
4 {
5     public static boolean isPalindrome(String i, int
6     {
7         if (low >= high) {
8             return true;
9         }
10
11         if (i.charAt(low) != i.charAt(high)) {
12             return false;
13         }
14
15         return isPalindrome(i, low + 1, high - 1);
16     }
17
18     public static void main(String[] args)
19     {
20         String i = "sravan";
21
22         if (isPalindrome(i, 0, i.length() - 1)) {
23             System.out.print("Yes");
24         } else {
25             System.out.print("No");
26         }
27     }
28 }
```

✕ Terminal



```
No  
Process finished.  
|
```



```
1  import java.io.*;
2
3
4
5  class Sravan
6  {
7
8
9  // Function to count digits
10
11  static int countEvenOdd(int n)
12  {
13
14      int even_count = 0;
15
16      int odd_count = 0;
17
18      while (n > 0)
19      {
20
21          int rem = n % 10;
22
23          if (rem % 2 == 0)
24              even_count++;
25
26          else
27              odd_count++;
28
29          n = n / 10;
30      }
31
32      System.out.println ( "Even count : " +
33                          even_count);
34
35      System.out.println ( "Odd count : " +
36                          odd_count);
37
38      if (even_count % 2 == 0 &&
39          odd_count % 2 != 0)
```



```
36     System.out.println ( "Even count : " +
37
38         even_count);
39
40     System.out.println ( "Odd count : " +
41
42         odd_count);
43
44     if (even_count % 2 == 0 &&
45         odd_count % 2 != 0)
46         return 1;
47
48     else
49         return 0;
50 }
51
52
53
54
55
56
57 // Driver Code
58
59 public static void main (String[] args)
60 {
61
62     int n;
63
64     n = 134732;
65
66     int t = countEvenOdd(n);
67
68
69
70
71     if (t == 1)
72
73         System.out.println ( "YES" );
74
75     else
76
77         System.out.println( "NO" ) ;
78
79
80
81 }
```





```
Even count : 2  
Odd count : 4  
NO  
  
Process finished.
```

```
3
4 import java.util.Arrays;
5
6
7
8 public class GFG
9 {
10
11     // A function to implement bubble sort
12
13     static void bubbleSort(int arr[], int n)
14     {
15
16         // Base case
17
18         if (n == 1)
19             return;
20
21
22         // One pass of bubble sort. After
23         // this pass, the largest element
24         // is moved (or bubbled) to end.
25         for (int i=0; i<n-1; i++)
26             if (arr[i] > arr[i+1])
27             {
28                 // swap arr[i], arr[i+1]
29
30                 int temp = arr[i];
31                 arr[i] = arr[i+1];
32                 arr[i+1] = temp;
33             }
34
35
36
37
38
39
40
41
42
43
44
45
46
```

```
39         int temp = arr[i];
40
41         arr[i] = arr[i+1];
42
43         arr[i+1] = temp;
44
45     }
46
47
48
49     // Largest element is fixed,
50
51     // recur for remaining array
52
53     bubbleSort(arr, n-1);
54
55 }
56
57
58
59 // Driver Method
60
61 public static void main(String[] args)
62
63 {
64
65     int arr[] = {64, 34, 25, 12, 22, 11, 90};
66
67
68
69     bubbleSort(arr, arr.length);
70
71
72
73     System.out.println("Sorted array : ");
74
75     System.out.println(Arrays.toString(arr));
76
77 }
78 }
```



Sorted array :

[11, 12, 22, 25, 34, 64, 90]

Process finished.

