

Form the largest number

ex:-

2 a b

"8"	"9"
-----	-----

"8" + "9" = "89"
"9" + "8" = "98"

$\left. \begin{array}{l} \text{"8 9"} = \\ \text{"9 8"} = \end{array} \right\} \begin{array}{l} \underline{a+b} \\ \underline{b+a} \end{array} \text{ concatenate}$

~~int~~

4	46	8	9
---	----	---	---

~~String~~

"4"	"46"	"8"	"9"
-----	------	-----	-----

~~sort~~

"9"	"8"	"46"	"4"
-----	-----	------	-----

~~answer~~

answer = "98464";

```
public static void largestNumber(int[] arr, int n) {
    String[] str = new String[arr.length];
    for (int i = 0; i < n; i++) {
        str[i] = String.valueOf(arr[i]);    // used to convert int to string
    }

    Arrays.sort(str, new myComparator());

    if (str[0].equals("0")) {
        System.out.println("0");
        return;
    }

    String answer = new String();    // declare String
    for (int i = 0; i < n; i++) {
        answer += str[i];
    }

    System.out.println(answer);
}

public static class myComparator implements Comparator<String> {
    @Override
    public int compare(String a, String b) {
        String case1 = a + b;
        String case2 = b + a;
        return case2.compareTo(case1);    // compareTo used to compare 2 values
    }
}
```

Max Subarray 2

5

-1	2	3	-2	1
----	---	---	----	---

↪ -1 = (-1)

↪ -1 2 = (1)

↪ -1 2 3 = (4)

↪ -1 2 3 -2 = (2)

↪ -1 2 3 -2 1 = (3)

↪ 2 = (2)

↪ 2 3 = (5)

↪ 2 3 -2 = (3)

↪ 2 3 -2 1 = (4)

↪ 3 = (3)

↪ 3 -2 = (1)

↪ 3 -2 1 = (2)

↪ -2 = (-2)

↪ -2 1 = (-1)

↪ 1 = (1)

```

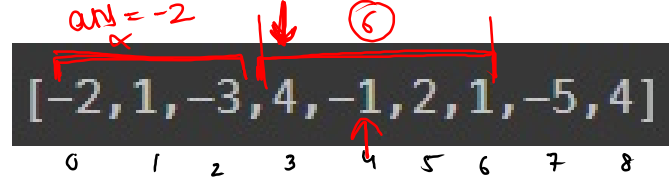
public static void maxSum(int[] arr, int n) {
    int ans = 0;
    int maxAns = Integer.MIN_VALUE;
    for (int i = 0; i < n; i++) {
        if (ans < 0) {
            ans = arr[i];
        } else {
            ans = ans + arr[i];
        }

        if (maxAns < ans) {
            maxAns = ans;
        }
    }

    System.out.println(maxAns);
}

```

nums = [5,4,-1,7,8]



ans = ~~0~~ ~~-2~~ ~~1~~ ~~-2~~ ~~6~~ ~~3~~ ~~4~~ ~~6~~ ~~5~~

maxAns = ~~-∞~~ ~~-2~~ ~~4~~ ~~6~~

Kadane's algorithm

