Note :-

top element of stack will always represents the previous potential element (which may make pair).

Postfix expression calculation

infix exp:- ((4+5)*(7-6)) // a+b

post exp:- 45+76-* // ab+

prefix exp:- *+45-76 // +ab

ans = 9

Calculate ans

str = "45+76-*"

0 1 2 3 4 5 6

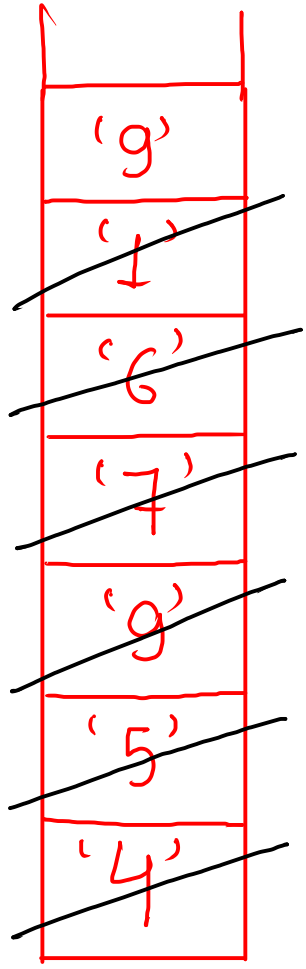
curr

top1 = ~~5~~ ~~6~~ 1

top2 = ~~4~~ ~~7~~ 9

ans = top2 * top1

= 9



stack

Integer

psudo code

1) create stack
2) traverse in string
    2.1) if curr ele. is number
        push curr ele.
    2.2) else
        curr ele = + , - , / , *
        top1
        top2
        calculate ans
        push ans in stack
3) return st.peek();

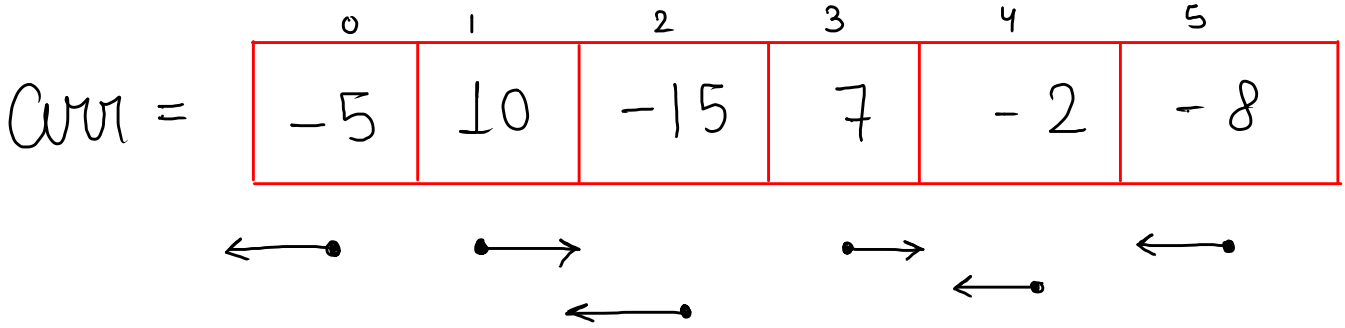
Code

```java
public static int postfixCalculate(String str) {
    Stack<Integer> st = new Stack<>();
    for (int i = 0; i < str.length(); i++) {
        char curr = str.charAt(i);
        if ( Character.isDigit(curr) ) {
            st.push( curr - '0' );
        } else {
            int top1 = st.peek();
            st.pop();
            int top2 = st.peek();
            st.pop();
            int ans = 0;
            if ( curr == '+' ) {
                ans = top2 + top1;
            } else if ( curr == '-' ) {
                ans = top2 - top1;
            } else if ( curr == '*' ) {
                ans = top2 * top1;
            } else if ( curr == '/' ) {
                ans = top2 / top1;
            }
            st.push(ans);
        }
    }
    return st.peek();
}
```

T.C = O(n)

S.C = O(n)

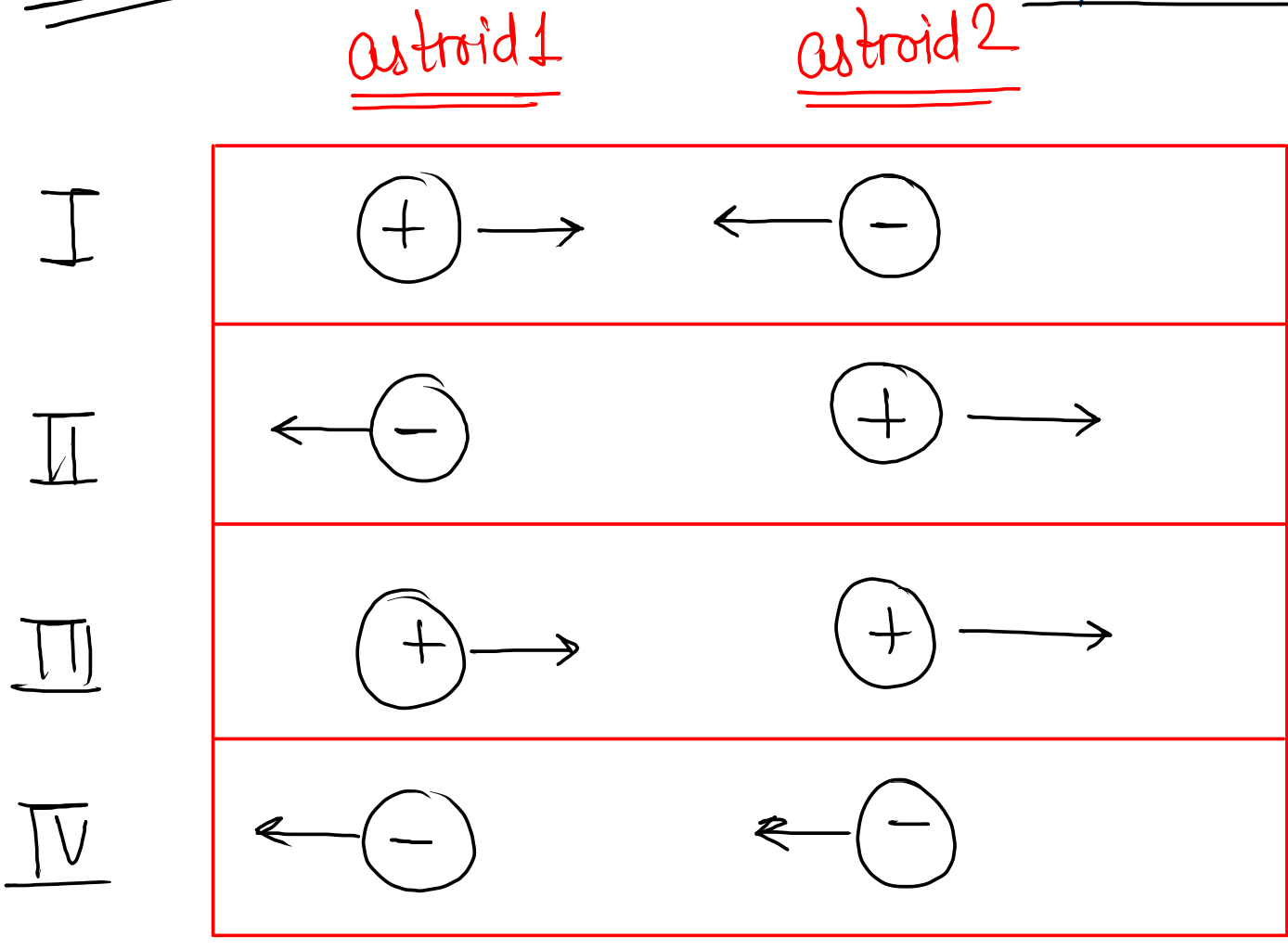
Asteroid Collision (M. 9mp)



Note:-

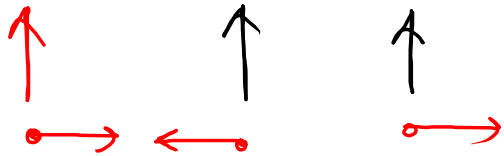
- absolute value of no. represents size of astroid
- + sign mean going in right
- - sign mean going in left

Observation (only collision case, curr ele is -ve, top ele is +ve)



Ex 1

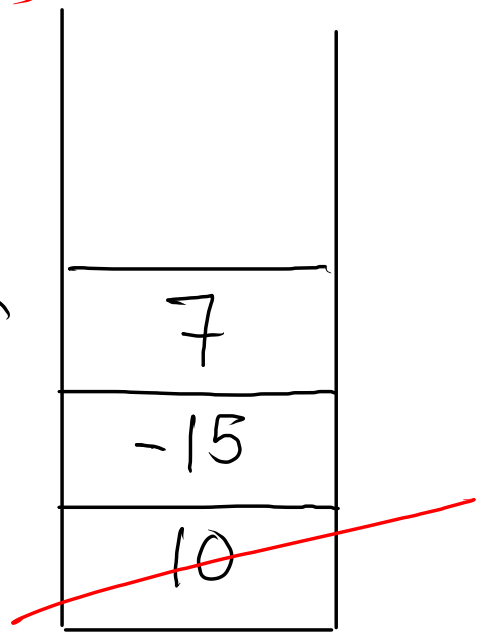
arr [10, -15, 7] = ans [-15, 7]



Curr ele = -ve

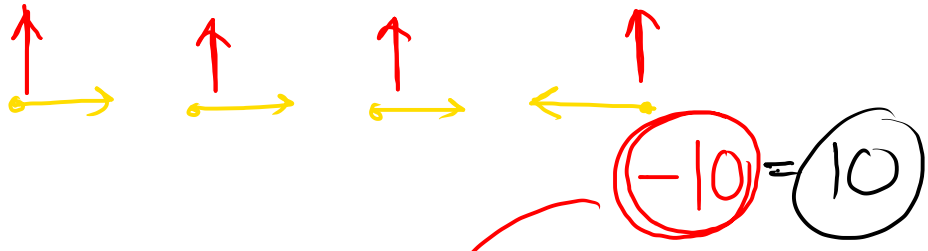
top ele = +ve

stack



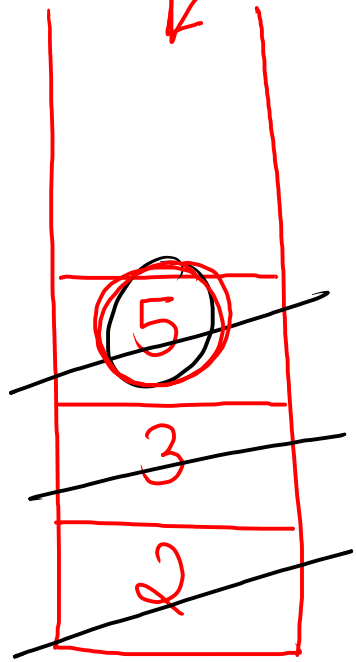
Ex2

arr [2 , 3 , 5 , -10]



curr = -ve

top = +ve




```
while(top < (curr * -1)){
    pop
}
```

psudo code

1) stack

2) loop in arr

2.1) if curr ele. is +ve

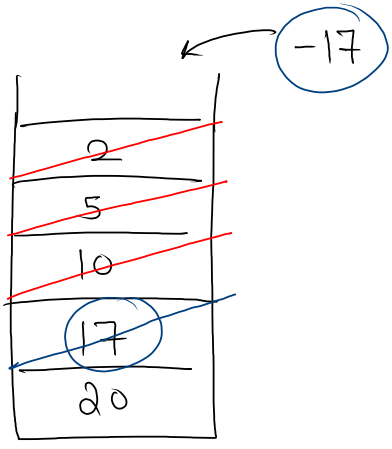
push curr ele.

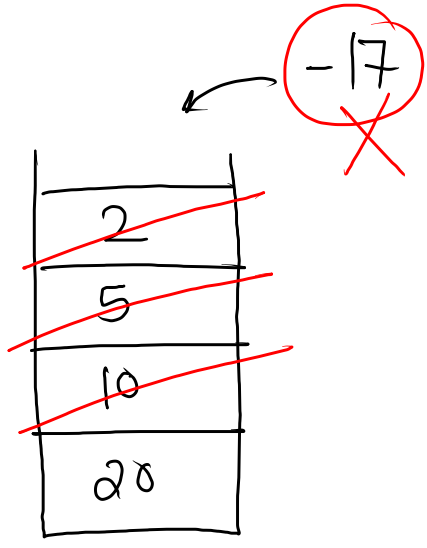
2.2) else

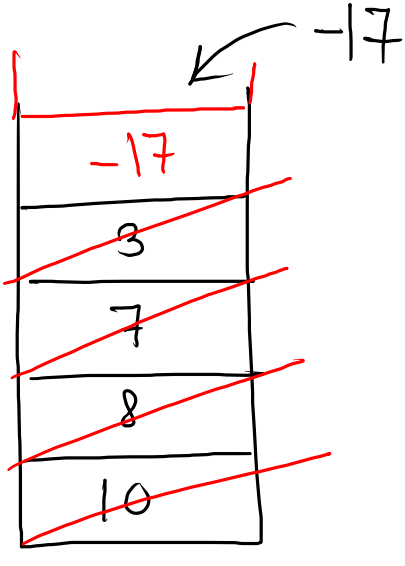
```
while(top < (curr * -1)){
    pop
}
```

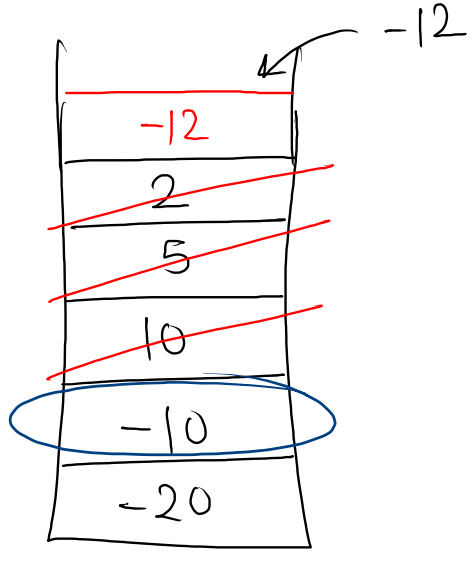
check if able to pop any element

Ex:-









Code

T.C = O(n), S.C = O(n)

```java
public static ArrayList<Integer> astroidCollision(int[] arr, int n) {
    Stack<Integer> st = new Stack<>();
    for (int i = 0; i < n; i++) {
        if ( arr[i] > 0 ) {
            st.push(arr[i]);
        } else {

            while ( st.size() > 0 && st.peek() > 0 && st.peek() < (-1 * arr[i]) ) {
                st.pop();
            }
            if ( st.size() > 0 && st.peek() == -1 * arr[i] ) {
                st.pop();
            } else if ( st.size() == 0 || st.peek() < 0) {
                st.push(arr[i]);
            }
        }
    }
    ArrayList<Integer> ans = new ArrayList<>();
    while ( st.size() > 0 ) {
        int top = st.peek();
        st.pop();

        ans.add(0, top);
    }
    return ans;
}
```