Q5 Move all the negative elements to the end. Also order of positive & negative elements should not change

i/p → {1, -1, 3, 2, -7, -5, 11, 6}
o/p → {1, 3, 2, 11, 6, -1, -7, -5}

In this we can create a temporary array & then we have to follow 3 steps.

1) Push the positive elements 1st in the temp array.
2) Then push the negative elements in the temp array.
3) Copy the values from temp array to the original array.

Dry run

1) i = 0
   temp → {1}
2) i = 1
   temp → {1}
3) i = 2
   temp → {1, 3}
4) i = 3
   temp → {1, 3, 2}
5) i = 4
   temp → {1, 3, 2}
6) i = 5
   temp → {1, 3, 2}
7) i = 6
   temp → {1, 3, 2, 11}

8) $i = 7$

temp → {1, 3, 2, 11, 6}

We are done with step-1 i.e copying positive elements in the temp array.

1) $i = 0$

temp → {1, 3, 2, 11, 6}

2) $i = 1$

temp → {1, 3, 2, 11, 6, -1}

3) $i = 2$

temp → {1, 3, 2, 11, 6, -1}

4) $i = 3$

temp → {1, 3, 2, 11, 6, -1}

5) $i = 4$

temp → {1, 3, 2, 11, 6, -1, -7}

6) $i = 5$

temp → {1, 3, 2, 11, 6, -1, -7, -5}

7) $i = 6$

temp → {1, 3, 2, 11, 6, -1, -7, -5}

8) $i = 7$

temp → {1, 3, 2, 11, 6, -1, -7, -5}

We are done with step-2 i.e copying the negative elements. Now the temp array is the answer & hence just copy it to original array.

arr → {1, 3, 2, 11, 6, -1, -7, -5}

Code

```
void segElements (vector<int> & arr) {

    vector<int> temp;
    for (int i=0; i<arr.size(); i++) {
            //Step-1 (Push +ve elements)
        if (arr[i] >=0) {
            temp.push_back(arr[i]);
        }
    }
    for (int i=0; i<arr.size(); i++) {
            //Step-2 (Push -ve elements)
        if (arr[i] <0) {
            temp.push_back(arr[i]);
        }
    }
    //Step-3 Copy temp to original array
    for (int i=0; i<temp.size(); i++) {
        arr[i] = temp[i];
    }
}
```

Time complexity = O(n)
Space complexity = O(n)