AP Computer Science Name_____
Manipulating Arrays Lab

OBJ: gain proficiency implementing one-dimensional arrays

Write a menu-driven program that performs the following tasks on a 1-D integer array of any length. Each task should be a separate function. Begin by asking for the users array size, and values for each position. No auxiliary arrays should be used except for Deleting all zeros where a new array may need to be created due to a change in size.

1. Display the array's length as well as its indices and elements.

Sample output

```
          size of array:      8
       array positions:       0   1   2   3   4   5   6   7
         array entries:      23 -10  17   0   0 -30   0   2
```

2. Search whether a value entered by the user is in the array. If found, it should give the first position it is found at; if not found, it should display an appropriate message.

Sample output

```
            What entry?    42
                status:    not found
    another search (y/n)?  y
            What entry?    17
                status:    found at position 2
    another search (y/n)?  n
```

3. Find the smallest element and exchange it with the first element. Display the array.

Sample output

```
   smallest element first
       array positions:       0   1   2   3   4   5   6   7
         array entries:     -30 -10  17   0   0  23   0   2
```

4. Rotate the array by a given number of steps. A positive number of steps rotates the array forward; a negative number of steps, backward.

Sample output

```
          how many steps?    -2
            rotated array
       array positions:       0   1   2   3   4   5   6   7
         array entries:      17   0   0  23   0   2 -30 -10
    rotate again (y/n)?     y
        how many steps?     10
            rotated array
       array positions:       0   1   2   3   4   5   6   7
         array entries:     -30 -10 17   0   0  23   0   2
    rotate again (y/n)?     n
```

5. Delete all zero elements from the current array, updating the length.

Sample output

```
          size of array:    5
       array positions:       0   1   2   3   4
         array entries:     -30 -10  17  23   2
```