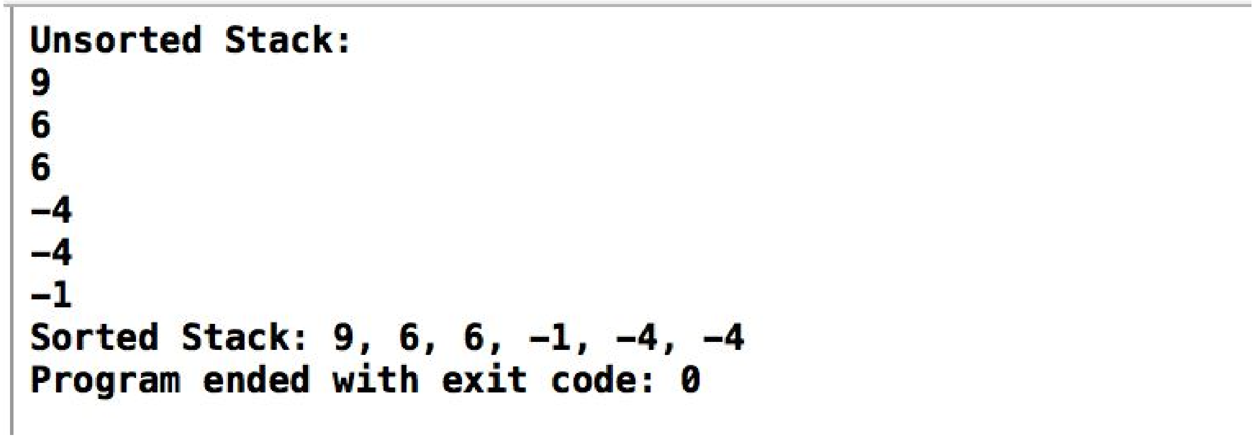
1.

a. Algorithm to sort a Stack in ascending order

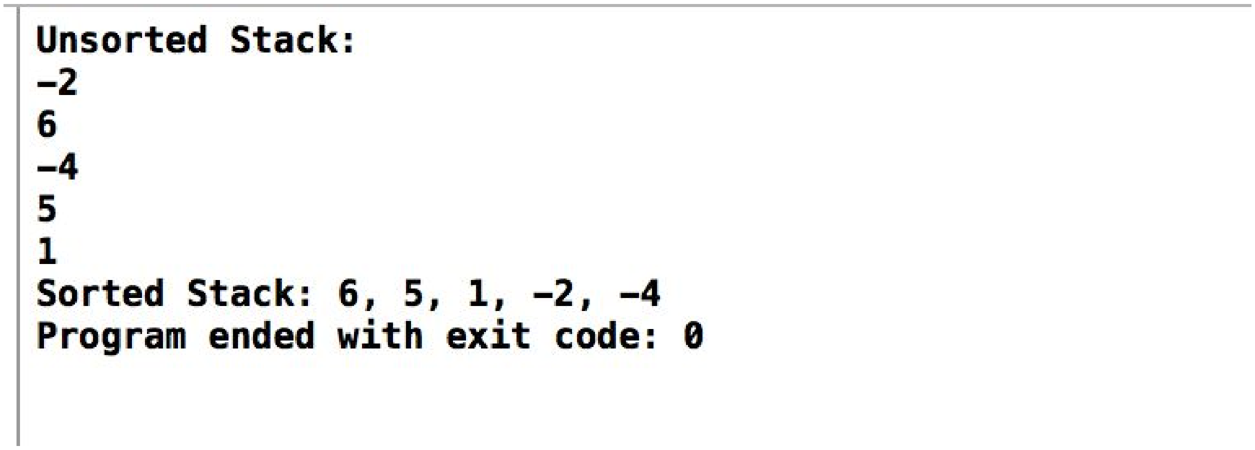
1. push elements onto stack one with the method push
2. Loop stack one while it is not empty
    a. Store the top element in stack one in a temporary variable. with the method top
    b. pop the top of the stack with the method pop
    c. check if the temporary variable is greater than or equal to the top in the second stack
    d. if it is NOT loop through the second stack while it is not empty
        i. push onto stack one, the top of stack two
        ii. pop stack two
    e. push temporary variable onto stack two.

b.

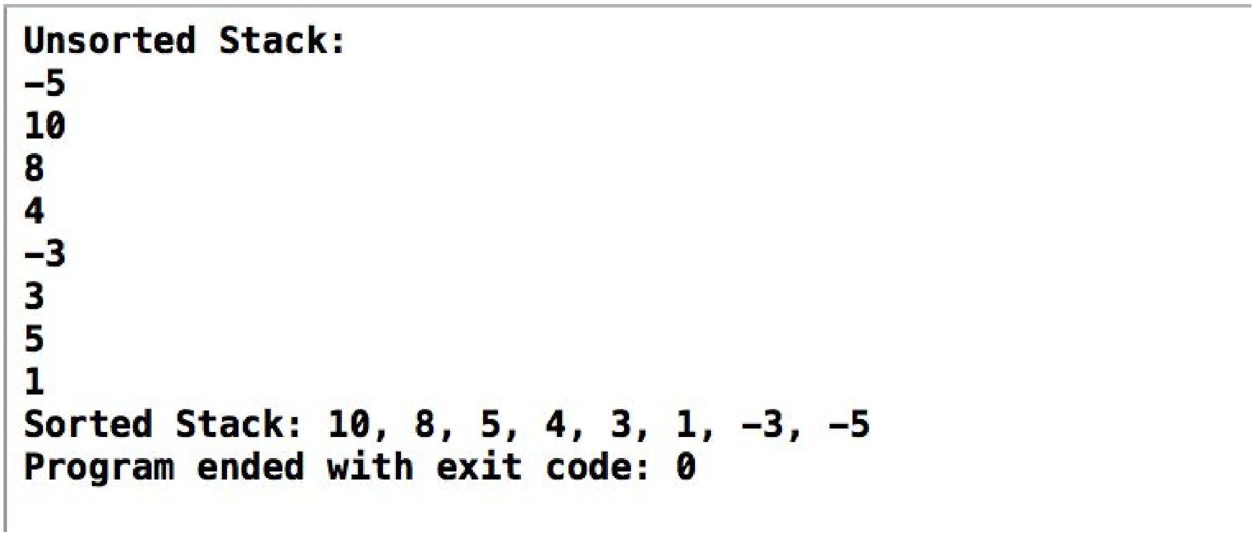
The few screen shots below show the functionality from part a in action. The output demonstrates the program is correct by first showing the stack unsorted, then showing it as assorted in ascending order (the top of the stack being the largest).



```
Unsorted Stack:
9
6
6
-4
-4
-1
Sorted Stack: 9, 6, 6, -1, -4, -4
Program ended with exit code: 0
```

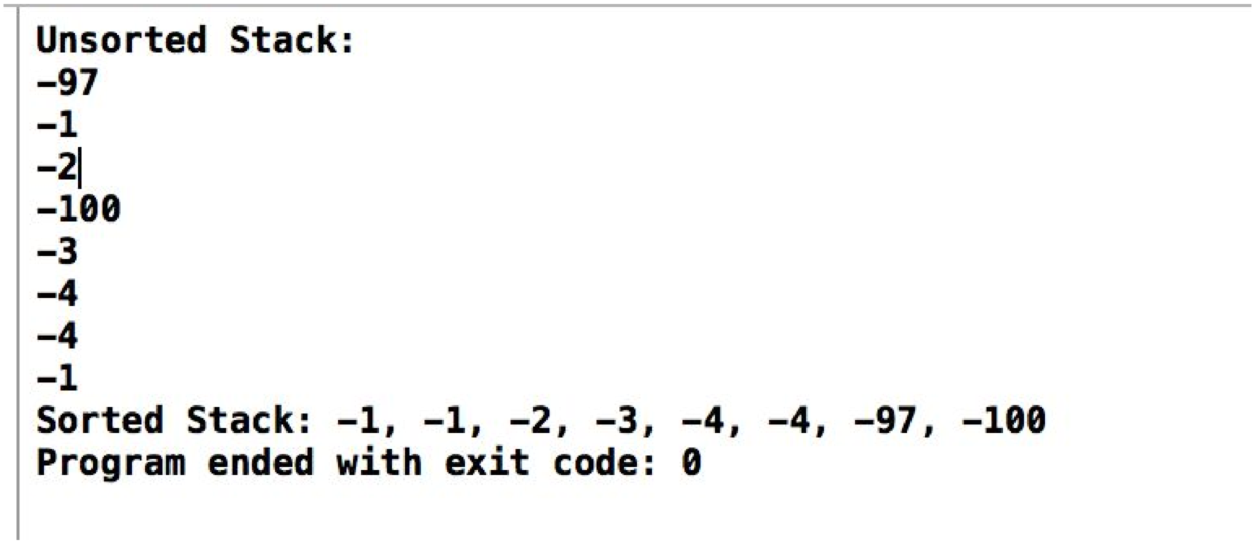


```
Unsorted Stack:
-2
6
-4
5
1
Sorted Stack: 6, 5, 1, -2, -4
Program ended with exit code: 0
```



```
Unsorted Stack:
-5
10
8
4
-3
3
5
1
Sorted Stack: 10, 8, 5, 4, 3, 1, -3, -5
Program ended with exit code: 0
```

All negatives



```
Unsorted Stack:
-97
-1
-2
-100
-3
-4
-4
-1
Sorted Stack: -1, -1, -2, -3, -4, -4, -97, -100
Program ended with exit code: 0
```