In [1]:
```
# Assignment 7
```

In [2]:
```
def second_smallest(input_list):
    if len(input_list) < 2:
        return None  # Not enough elements to find the second smallest

    smallest = min(input_list)
    second_smallest = float('inf')  # Initialize with positive infinity

    for num in input_list:
        if num > smallest and num < second_smallest:
            second_smallest = num

    return second_smallest
```

In [3]:
```
# Example usage
numbers = [1, 2, -8, -2, 0]
result = second_smallest(numbers)

if result is not None:
    print("Second smallest number:", result)
else:
    print("There is no second smallest number in the list.")
```

```
Second smallest number: -2
```

In [4]:
```
def swap_first_last_chars(input_string):
    if len(input_string) < 2:
        return input_string  # No change needed for strings with length less than 2

    first_char = input_string[0]
    last_char = input_string[-1]
    middle_chars = input_string[1:-1]

    new_string = last_char + middle_chars + first_char
    return new_string
```

In [6]:
```
original_string = "hello"
new_string = swap_first_last_chars(original_string)
print("Original string:", original_string)
print("New string:", new_string)
```

```
Original string: hello
New string: oellh
```

In [9]:
```
def find_longest_word_length(word_list):
    longest_length = 0
    for word in word_list:
        if len(word) > longest_length:
            longest_length = len(word)
    return longest_length
```

In [10]:
```
word_list = ["apple", "banana", "grapefruit", "kiwi", "watermelon"]
longest_length = find_longest_word_length(word_list)
print("Length of the longest word:", longest_length)
```

```
Length of the longest word: 10
```

In [11]:
```
def remove_nth_character(input_string, n):
    if n < 0 or n >= len(input_string):
        return "Invalid index"

    new_string = input_string[:n] + input_string[n+1:]
    return new_string
```

In [13]:
```
original_string = "hello"
index_to_remove = 2
new_string = remove_nth_character(original_string, index_to_remove)

print("Original string:", original_string)
print("String after removing character at index", index_to_remove, ":", new_string)
```

```
Original string: hello
String after removing character at index 2 : helo
```

In [14]:
```
def key_exists(dictionary, key):
    return key in dictionary
```

In [18]:
```
# Example dictionary
d = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

# Check if keys exist
key_to_check = '5'
if key_exists(d, key_to_check):
    print(f"The key '{key_to_check}' exists in the dictionary.")
else:
    print(f"The key '{key_to_check}' does not exist in the dictionary.")
```

```
The key '5' does not exist in the dictionary.
```

In [ ]:
```

```