1. Write a Python program to find the second smallest number in a list.

input second_smallest([1, 2, -8, -2, 0]) output -2

In [2]:
```python
def second_smallest(numbers):
    smallest = float('inf')
    second_smallest = float('inf')

    for num in numbers:
        if num < smallest:
            second_smallest = smallest
            smallest = num
        elif num < second_smallest and num != smallest:
            second_smallest = num

    return second_smallest

numbers = [1, 2, -8, -2, 0]
result = second_smallest(numbers)
print(result)
```

```
-2
```

2. Write a Python program to change a given string to a new string where the first and last chars have been exchanged

In [3]:
```python
def exchange_first_last(string):
    if len(string) <= 1:
        return string
    else:
        first_char = string[0]
        last_char = string[-1]
        middle_chars = string[1:-1]
        return last_char + middle_chars + first_char

input_string = "Hello, World!"
new_string = exchange_first_last(input_string)
print(new_string)
```

```
!ello, WorldH
```

3. Write a Python function that takes a list of words and returns the length of the longest one

In [4]:
```python
def find_longest_word(words):
    longest_length = 0

    for word in words:
        if len(word) > longest_length:
            longest_length = len(word)

    return longest_length

word_list = ['apple', 'banana', 'cherry', 'dragonfruit', 'elderberry']
longest_length = find_longest_word(word_list)
print(longest_length)
```

```
11
```

4. Write a Python program to remove the nth index character from a nonempty string

In [5]:
```python
def remove_nth_character(string, n):
    if n < 0 or n >= len(string):
        return string
    else:
        return string[:n] + string[n+1:]

input_string = "Hello, World!"
index_to_remove = 7
new_string = remove_nth_character(input_string, index_to_remove)
print(new_string)
```

```
Hello, orld!
```

5. Check if a given key already exists in a dictionary

input d = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60} is_key_present(5) is_key_present(9) output Key is present in the dictionary
Key is not present in the dictionary

In [6]:
```
def is_key_present(dictionary, key):
    if key in dictionary:
        return "Key is present in the dictionary"
    else:
        return "Key is not present in the dictionary"

d = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

print(is_key_present(d, 5))
print(is_key_present(d, 9))
```

```
Key is present in the dictionary
Key is not present in the dictionary
```

In [ ]: