Write a Python program to find the second smallest number in a list.
input
second_smallest([1, 2, -8, -2, 0])
output
-2

In [1]:
```
def second_smallest(numbers):
    if len(numbers) < 2:
        return None

    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

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

```
-2
```

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

In [2]:
```
def exchange_first_last_chars(string):
    if len(string) < 2:
        return string

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

    new_string = last_char + middle_chars + first_char
    return new_string

# Example usage
string = "Hello"
new_string = exchange_first_last_chars(string)
print(new_string)
```

```
oellH
```

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

In [3]:
```
def find_longest_word_length(word_list):
    longest_length = 0

    for word in word_list:
        current_length = len(word)
        if current_length > longest_length:
            longest_length = current_length

    return longest_length

# Example usage
words = ["apple", "banana", "cherry", "durian"]
longest_length = find_longest_word_length(words)
print(longest_length)
```

```
6
```

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

In [4]:
```
def remove_nth_character(string, n):
    if n < 0 or n >= len(string):
        return string

    new_string = string[:n] + string[n+1:]
    return new_string

# Example usage
string = "Hello, World!"
n = 7
new_string = remove_nth_character(string, n)
print(new_string)
```

```
Hello, orld!
```

```
# 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 [5]:
```
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"

# Example usage
d = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
key1 = 5
key2 = 9

result1 = is_key_present(d, key1)
result2 = is_key_present(d, key2)

print(result1)
print(result2)
```

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

In [ ]: