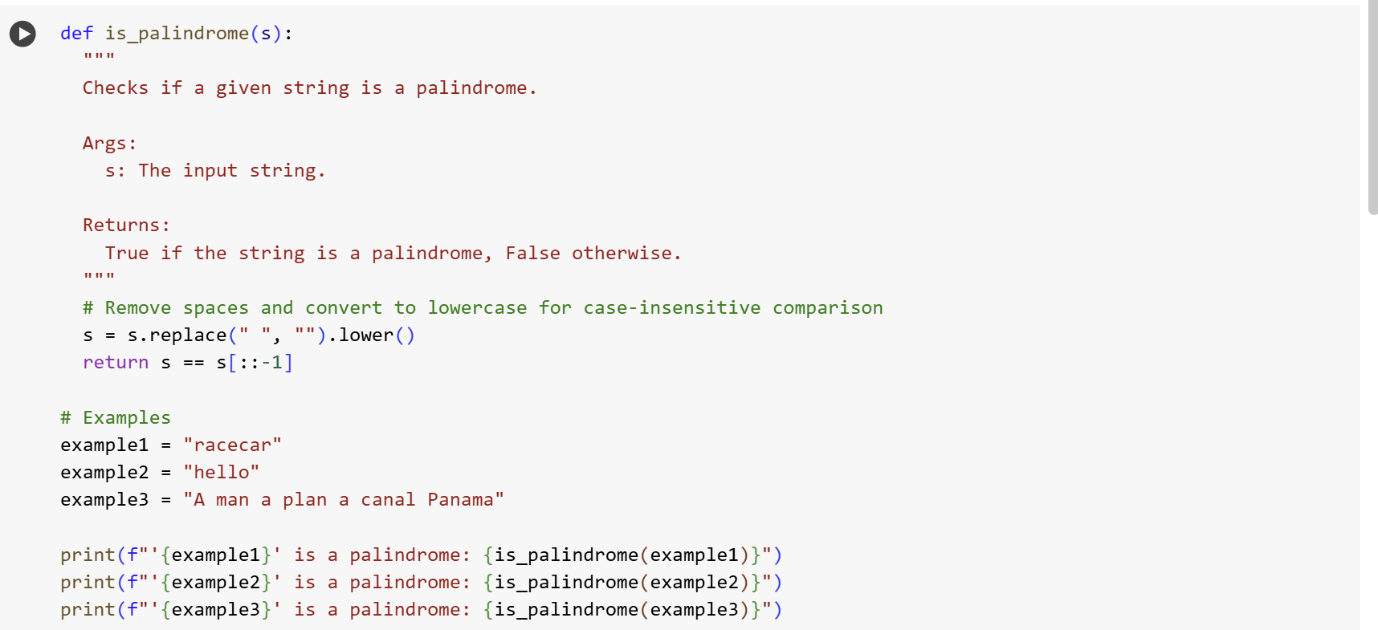
HTNO:2403A52134

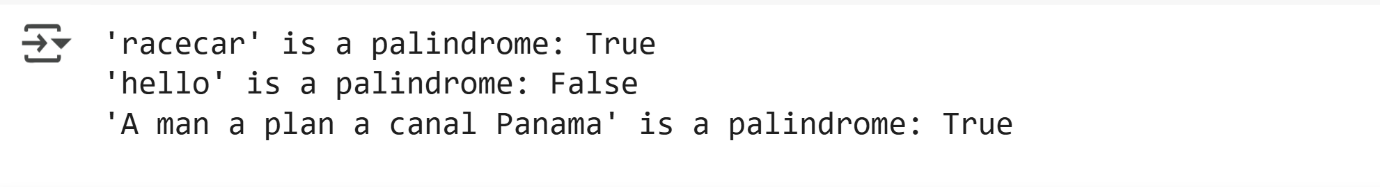
Lab Assessment-1:

Q1:To check whether the given string is palindrome or not.

Code :



Output:

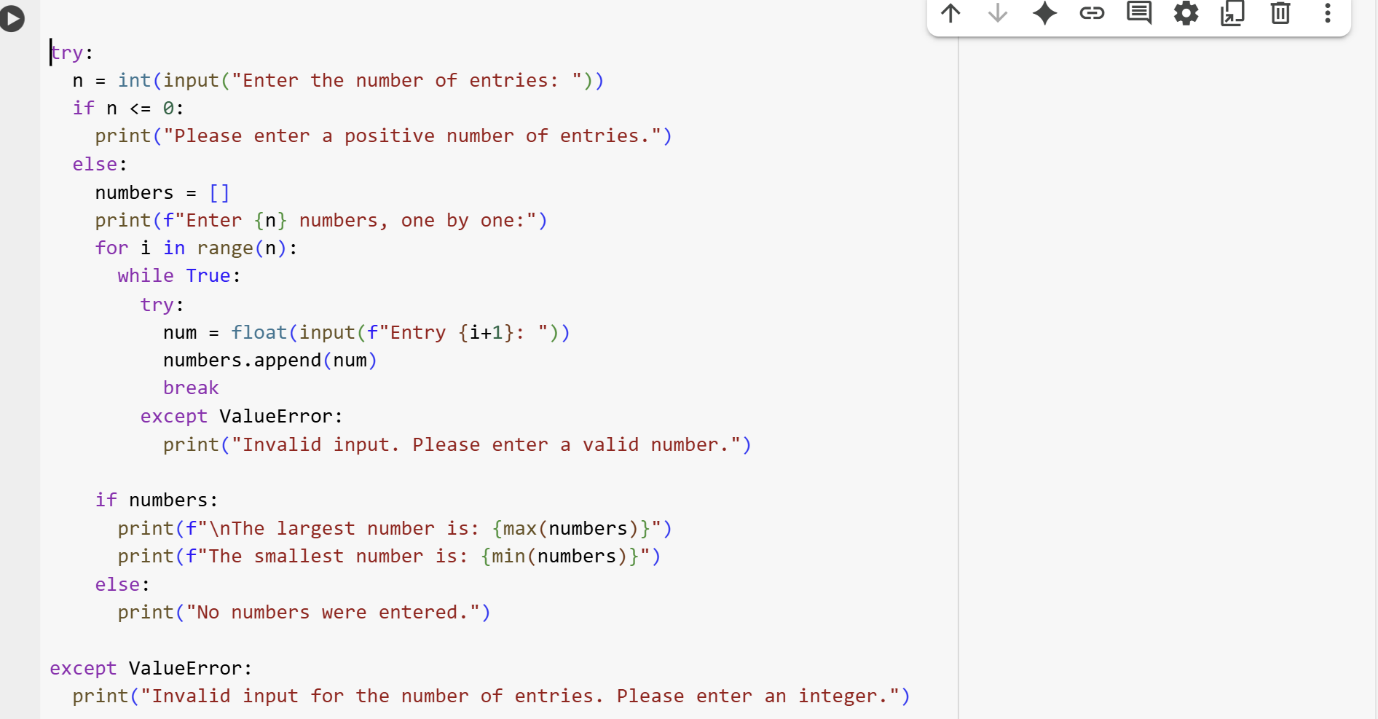


Explanation:

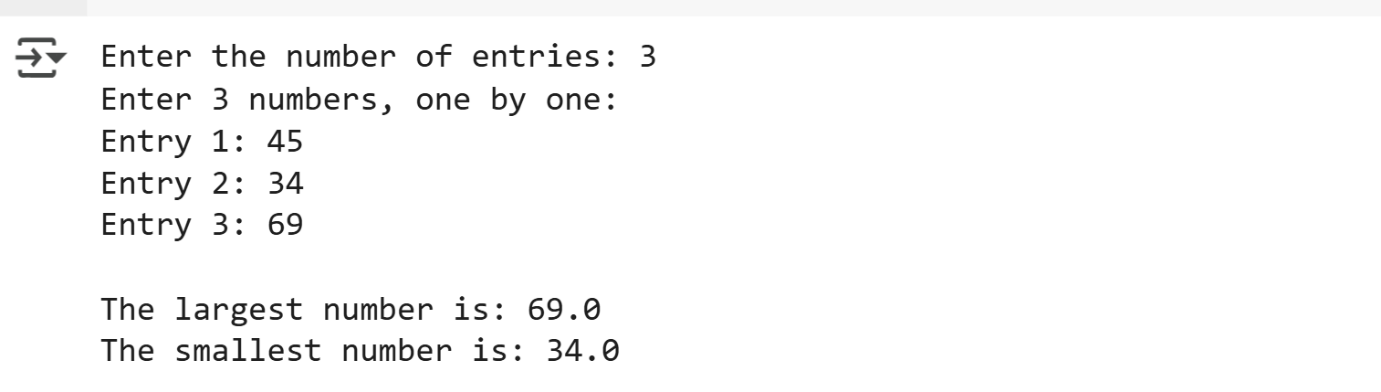


Q2:To generate the largest and smallest number in the user provided list

Code:



Output:



Explanation:

* try:: This starts a try block, which is used for error handling. The code inside this block will be executed, and if an error occurs, the code in the except block will be executed.
* n = int(input("Enter the number of entries: ")): This line prompts the user to enter the number of entries and stores it as an integer in the variable n.
* if n <= 0:: This checks if the entered number of entries is less than or equal to zero.
* print("Please enter a positive number of entries."): If n is not positive, this message is printed.
* else:: If n is a positive number, the code in this else block is executed.
* numbers = []: An empty list called numbers is created to store the user's input.
* print(f"Enter {n} numbers, one by one:"): This line tells the user to enter the numbers.
* for i in range(n):: This loop iterates n times, once for each entry.
* while True:: This starts an inner loop that continues until a valid number is entered.
* try:: This inner try block handles potential errors when converting the user's input to a number.
* num = float(input(f"Entry {i+1}: ")): This prompts the user to enter a number for the current entry and converts it to a float, storing it in the num variable.
* numbers.append(num): The entered number (num) is added to the numbers list.
* break:: If the number is successfully entered and added to the list, this statement exits the inner while loop.
* except ValueError:: If the user's input cannot be converted to a float, this except block is executed.
* print("Invalid input. Please enter a valid number."): This message is printed if the input is not a valid number, and the inner while loop continues, asking for input again.
* if numbers:: After the loop finishes, this checks if the numbers list is not empty.
* print(f"\nThe largest number is: {max(numbers)}"): If there are numbers in the list, this line finds the largest number using max() and prints it.
* print(f"The smallest number is: {min(numbers)}"): This line finds the smallest number using min() and prints it.
* else:: If the numbers list is empty (because the user entered 0 or a negative number for n), this else block is executed.
* print("No numbers were entered."): This message is printed if no numbers were entered try:: This starts a try block, which is used for error handling. The code inside this block will be executed, and if an error occurs, the code in the except block will be executed.
* n = int(input("Enter the number of entries: ")): This line prompts the user to enter the number of entries and stores it as an integer in the variable n.
* if n <= 0:: This checks if the entered number of entries is less than or equal to zero.
* print("Please enter a positive number of entries."): If n is not positive, this message is printed.
* else:: If n is a positive number, the code in this else block is executed.
* numbers = []: An empty list called numbers is created to store the user's input.
* print(f"Enter {n} numbers, one by one:"): This line tells the user to enter the numbers.
* for i in range(n):: This loop iterates n times, once for each entry.
* while True:: This starts an inner loop that continues until a valid number is entered.
* try:: This inner try block handles potential errors when converting the user's input to a number.
* num = float(input(f"Entry {i+1}: ")): This prompts the user to enter a number for the current entry and converts it to a float, storing it in the num variable.
* numbers.append(num): The entered number (num) is added to the numbers list.
* break:: If the number is successfully entered and added to the list, this statement exits the inner while loop.
* except ValueError:: If the user's input cannot be converted to a float, this except block is executed.
* print("Invalid input. Please enter a valid number."): This message is printed if the input is not a valid number, and the inner while loop continues, asking for input again.
* if numbers:: After the loop finishes, this checks if the numbers list is not empty.
* print(f"\nThe largest number is: {max(numbers)}"): If there are numbers in the list, this line finds the largest number using max() and prints it.
* print(f"The smallest number is: {min(numbers)}"): This line finds the smallest number using min() and prints it.
* else:: If the numbers list is empty (because the user entered 0 or a negative number for n), this else block is executed.
* print("No numbers were entered."): This message is printed if no numbers were entered.
* except ValueError:: This except block handles the case where the initial input for the number of entries (n) is not a valid integer.
* print("Invalid input for the number of entries. Please enter an integer."): This message is printed if the initial input for n is invalid.
* except ValueError:: This except block handles the case where the initial input for the number of entries (n) is not a valid integer.
* print("Invalid input for the number of entries. Please enter an integer."): This message is printed if the initial input for n is invalid.