

Code:

```
GNU nano 6.2 largest_product.py
#!/usr/bin/env python3

def largest_series_product(input_string, series_length):
    try:
        # Input validation
        if not input_string.isdigit():
            raise ValueError("digits input must only contain digits") # Series contains non-numeric input

        if not isinstance(series_length, int):
            raise ValueError("length must be an integer") # Series length must be an integer

        if series_length < 0:
            raise ValueError("length must not be negative") # Series length cannot be negative

        if series_length > len(input_string):
            raise ValueError("length must be smaller than string length") # Series length too long

        max_product = 0 # To track the largest product
        max_series = "" # To track the series that gives the largest product

        # Iterate through possible series
        for i in range(len(input_string) - series_length + 1):
            series = input_string[i:i + series_length] # Extract series
            product = 1
            for digit in series:
                product *= int(digit) # Calculate the product of the series

            # Update max_product and max_series if a larger product is found
            if product > max_product:
                max_product = product
                max_series = series

        return f"Largest series product is {max_product} from the series '{max_series}'."

    except ValueError as e:
        return f"Error: {e}"

if __name__ == "__main__":
    # Get user input for the string and length
    user_input_string = input("Enter the sequence of digits: ")
    try:
        user_length = int(input("Enter the length of the series: ")) # Get the length as an integer
        # Call the function with user input
        result = largest_series_product(user_input_string, user_length)
        print(result) # Print the result
    except ValueError as e:
        print(f"Error: {e}") # Handle non-integer length input or any other errors
```

Results:

```
vboxuser@Cloudia:~$ ./largest_product.py
Enter the sequence of digits: 874990898
Enter the length of the series: 4
Largest series product is 2268 from the series '7499'.
vboxuser@Cloudia:~$ ./largest_product.py
Enter the sequence of digits: 28387983h8
Enter the length of the series: 2
Error: digits input must only contain digits
vboxuser@Cloudia:~$ ./largest_product.py
Enter the sequence of digits: 1238489
Enter the length of the series: -3
Error: length must not be negative
vboxuser@Cloudia:~$ ./largest_product.py
Enter the sequence of digits: 23
Enter the length of the series: 3
Error: length must be smaller than string length
```