

COMPUTATIONAL THINKING TRANSCRIPT

Strong Number Checker

User

give me a code to find if a given input number is a strong number. A strong number is a number whose sum of factorials of digits is equal to the number itself

ChatGPT

Sure, here's a Python code to check if a given input number is a strong number:

python

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

def is_strong_number(num):
    temp = num
    sum_of_factorials = 0

    while temp > 0:
        digit = temp % 10
        sum_of_factorials += factorial(digit)
        temp //= 10

    return sum_of_factorials == num

# Test the function
number = int(input("Enter a number to check if it's a strong number: "))
if is_strong_number(number):
    print(number, "is a strong number.")
else:
    print(number, "is not a strong number.")
```

This code defines two functions: `factorial()` to calculate the factorial of a number and `is_strong_number()` to check if a given number is a strong number or not. Finally, it takes user input and checks whether the input number is a strong number or not and prints the result accordingly.

User

divide all the small tasks into separate reusable functions in the above code

ChatGPT

Sure, let's divide the code into separate reusable functions:

python

```
def factorial(n):
    """Calculate the factorial of a number."""
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

def sum_of_factorials_of_digits(num):
    """Calculate the sum of factorials of digits of a number."""
    temp = num
    sum_of_factorials = 0

    while temp > 0:
        digit = temp % 10
        sum_of_factorials += factorial(digit)
        temp //= 10

    return sum_of_factorials

def is_strong_number(num):
    """Check if a given number is a strong number."""
    return sum_of_factorials_of_digits(num) == num

def main():
    """Main function to take user input and check if it's a strong number."""
```

```

number = int(input("Enter a number to check if it's a strong number: "))
if is_strong_number(number):
    print(number, "is a strong number.")
else:
    print(number, "is not a strong number.")

# Call the main function
if __name__ == "__main__":
    main()

```

In this revised version, each logical part of the code is encapsulated within a separate function with a clear purpose. This makes the code more modular and reusable. The main() function is used to orchestrate the entire process of taking user input and checking if it's a strong number.

User

can you divide the sum of factorials of digits further into smaller functions
ChatGPT

Certainly! Let's divide the sum_of_factorials_of_digits function further into smaller functions:

```

python
def calculate_factorial(digit):
    """Calculate the factorial of a digit."""
    if digit == 0:
        return 1
    else:
        result = 1
        for i in range(1, digit + 1):
            result *= i
        return result

def sum_of_factorials_of_digits(num):
    """Calculate the sum of factorials of digits of a number."""
    def get_last_digit(n):
        return n % 10

```

```

def remove_last_digit(n):
    return n // 10

temp = num
sum_of_factorials = 0

while temp > 0:
    digit = get_last_digit(temp)
    sum_of_factorials += calculate_factorial(digit)
    temp = remove_last_digit(temp)

return sum_of_factorials

# The rest of the code remains the same as before

def is_strong_number(num):
    """Check if a given number is a strong number."""
    return sum_of_factorials_of_digits(num) == num

def main():
    """Main function to take user input and check if it's a strong number."""
    number = int(input("Enter a number to check if it's a strong number: "))
    if is_strong_number(number):
        print(number, "is a strong number.")
    else:
        print(number, "is not a strong number.")

# Call the main function
if __name__ == "__main__":
    main()

```

In this version, I've split the calculation of factorial into a separate function called `calculate_factorial`. Additionally, I've created two helper functions within the `sum_of_factorials_of_digits` function: `get_last_digit` to retrieve the last digit of a number and `remove_last_digit` to remove the

last digit from the number. This enhances readability and makes each function focused on a single task.

User

can you write those get last digit and remove last digit functions as independent functions

ChatGPT

Of course! Here are the `get_last_digit` and `remove_last_digit` functions as independent functions:

python

```
def get_last_digit(n):
    """Get the last digit of a number."""
    return n % 10

def remove_last_digit(n):
    """Remove the last digit from a number."""
    return n // 10
```

Now we can integrate these functions into the previous code.

```
def calculate_factorial(digit):
    """Calculate the factorial of a digit."""
    if digit == 0:
        return 1
    else:
        result = 1
        for i in range(1, digit + 1):
            result *= i
        return result

def sum_of_factorials_of_digits(num):
    """Calculate the sum of factorials of digits of a number."""
    temp = num
    sum_of_factorials = 0

    while temp > 0:
        digit = get_last_digit(temp)
```

```
    sum_of_factorials += calculate_factorial(digit)
    temp = remove_last_digit(temp)
```

```
    return sum_of_factorials
```

The rest of the code remains the same as before

```
def is_strong_number(num):
    """Check if a given number is a strong number."""
    return sum_of_factorials_of_digits(num) == num
```

```
def main():
    """Main function to take user input and check if it's a strong number."""
    number = int(input("Enter a number to check if it's a strong number: "))
    if is_strong_number(number):
        print(number, "is a strong number.")
    else:
        print(number, "is not a strong number.")
```

```
# Call the main function
if __name__ == "__main__":
    main()
```

Now, the `get_last_digit` and `remove_last_digit` functions are independent and can be reused in any part of the code.