def display_output():
    number = 1
    count = 10
    value = 4

    while count > 4:
        count = count - 2
        print(str(number) + ":", count, value)
        value = value + count
        number = number + 1

    print()
    print(str(number) + ":", count, value)

def main():
    display_output()

main()
Complete the function

The get_dice_throws_result() function throws a number of dice (given by num_throws) and counts how often the dice value, (given by dice_to_check) occurs. Complete the function.

```python
import random
def get_dice_throws_result(num_throws, dice_to_check):

def main():
    print("30000 throws," , get_dice_throws_result(30000, 6), "sixes")
    print("6 throws," , get_dice_throws_result(6, 6), "sixes")
    print("600000 throws," , get_dice_throws_result(600000, 5), "fives")

main()
```

30000 throws, 4913 sixes
6 throws, 0 sixes
600000 throws, 99929 fives
Complete the function

For an integer, a divisor is a number which divides exactly into the integer (a factor of the integer), e.g., the divisors of 6 are 1, 2, 3, 6. Note that 1 and the number itself are divisors (they divide into the number exactly). Complete the `string_of_divisors()` function.

```python
def string_of_divisors(number):
    pass

def main():
    string_of_divisors(24)
    string_of_divisors(25)
    string_of_divisors(5628)

main()
```

1 2 3 4 6 8 12 24
1 5 25
1 2 3 4 6 7 12 14 21 28 42 67 84 134 201 268 402 469 804 938 1407 1876 2814 5628
Complete the function

For an integer, a divisor is a number which divides exactly into the integer (a factor of the integer), e.g., the divisors of 6 are 1, 2, 3, 6. Note that 1 and the number itself are divisors (they divide into the number exactly). For this function we only want the sum of all the divisors less than the number itself. Complete the function.

```python
def get_sum_of_divisors(number):
    pass

def main():
    print("get_sum_of_divisors(6)", get_sum_of_divisors(6))
    print("get_sum_of_divisors(24)", get_sum_of_divisors(24))
    print("get_sum_of_divisors(25)", get_sum_of_divisors(25))
    print("get_sum_of_divisors(5628)", get_sum_of_divisors(5628))
main()
```
A perfect number is an integer that is equal to the sum of its divisors (including 1, excluding the number itself), e.g., the sum of the divisors of 28 is 28 (1 + 2 + 4 + 7 + 14). Complete the `check_perfection()` function which checks for perfection and prints either '# is a perfect number' or '# is NOT a perfect number'.

```python
def get_sum_of_divisors(number):
    #See the previous Slide

def check_perfection(number):
    message_is = "is a perfect number"
    message_is_not = "is NOT a perfect number"

def main():
    check_perfection(28)  # 28 is a perfect number
    check_perfection(54)  # 54 is NOT a perfect number
    check_perfection(496) # 496 is a perfect number

main()
```
Complete the function

Complete the user_number_guess() function which keeps prompting the user to guess a hidden number until the user correctly guesses the number. At each guess the function lets the user know if the guess is too high or too low. At the end, the function also prints the number of guesses taken.

```python
def user_number_guess(computer_num):
    prompt = "Enter your guess (1 - 99): "
    num_guesses = 0

    print(prompt, end='')
    guess = int(input())
    num_guesses += 1

    while guess != computer_num:
        if guess > computer_num:
            print("Too high")
        else:
            print("Too low")

        print(prompt, end='')
        guess = int(input())
        num_guesses += 1

    print("Correct! Number of guesses: ", num_guesses)

def main():
    user_number_guess(random.randrange(1, 100))

main()
```

Enter your guess (1 - 99): 50
Too high
Enter your guess (1 - 99): 25
Too high
Enter your guess (1 - 99): 13
Too low
Enter your guess (1 - 99): 20
Too low
Enter your guess (1 - 99): 23
Correct! Number of guesses: 5
Complete the function

The `get_legal_number()` function repeatedly prompts the user for a number until the user number is within (both inclusive) the two numbers passed as parameters. The function returns the user number. Complete the function.

```python
def get_legal_user_num(lower, upper):
    prompt = "Enter a number (" + str(lower) + " - " + str(upper) + "): 
    return int(input(prompt))

def main():
    print(get_legal_user_num(0, 6))
    print(get_legal_user_num(10, 20))
    print(get_legal_user_num(1, 2))

main()
```

Enter a number (0-6): -1
Enter a number (0-6): 8
Enter a number (0-6): 6
Enter a number (10-20): 13
Enter a number (1-2): 3
Enter a number (1-2): 0
Enter a number (1-2): 2

2