using python3, Create a program that takes (at least) three command line arguments. The first two will be integers and the third will be a float.\n\n On one line, separated by spaces, print the sum of the first two arguments, the product of the first and third arguments, the first argument modulo the second, and the integer quotient of the third by the first.\n Add 1 to all three arguments.\n On a new line, print the first argument bitwise right shift 3, the second argument divided by 2 (not integer division), and the bitwise OR of the first and second arguments. (all separated by spaces.)\n On the last line, print the sum of the first argument (after the addition) and the total number of arguments, excluding the program name

Versus Chatgpt response:

Here is the Python 3 code to accomplish the given task:

import sys

parse command line arguments

\narg1 = int(sys.argv[1])

\narg2 = int(sys.argv[2])

\narg3 = float(sys.argv[3])\n\n

part 1

\nprint(arg1 + arg2, arg1 * arg3, arg1 % arg2, int(arg3 / arg1))\n\n

add 1 to all arguments

\narg1 += 1\narg2 += 1\narg3 += 1\n\n

part 2

\nprint(arg1 >> 3, arg2 / 2, arg1 | arg2)\n\n

part 3\nprint(arg1 + len(sys.argv) - 1)\n\n

Note that this program assumes that the command line arguments are provided in the correct format and order. It is recommended to add error checking and handling to ensure the program runs correctly in all scenarios

Certainly! Below is a Python program that meets the specified requirements:

python
Copy

```
import sys

def main():
    # Check if at least three arguments are provided
    if len(sys.argv) < 4:
        print("Usage: python3 program_name.py <int1> <int2> <float1>")
        return

    # Extract the arguments
    int1 = int(sys.argv[1])
    int2 = int(sys.argv[2])
    float1 = float(sys.argv[3])

    # Perform the required calculations
    sum_of_first_two = int1 + int2
    product_of_first_and_third = int1 * float1
    modulo_of_first_by_second = int1 % int2
    quotient_of_third_by_first = int(float1 // int1)

    # Add 1 to all three arguments
    int1 += 1
    int2 += 1
    float1 += 1

    # Print the results on the first line
    print(f"{sum_of_first_two} {product_of_first_and_third} {modulo_of_first_by_second} {quotient_of_third_by_f
irst}")

    # Perform additional calculations
    right_shift_of_first = int1 >> 3
    division_of_second_by_2 = int2 / 2
    bitwise_or_of_first_and_second = int1 | int2

    # Print the results on the second line
    print(f"{right_shift_of_first} {division_of_second_by_2} {bitwise_or_of_first_and_second}")

    # Calculate the sum of the first argument (after addition) and the total number of arguments
    total_arguments = len(sys.argv) - 1
    sum_of_first_and_total_arguments = int1 + total_arguments

    # Print the result on the last line
    print(sum_of_first_and_total_arguments)

if __name__ == "__main__":
    main()
```

Explanation:

1. **Argument Parsing**:

- The program checks if at least three arguments are provided using `len(sys.argv)`.
- The first two arguments are converted to integers (`int1` and `int2`), and the third argument is converted to a float (`float1`).

2. **Calculations**:
   - The sum of the first two arguments is calculated.
   - The product of the first argument and the third argument is calculated.
   - The first argument modulo the second argument is calculated.
   - The integer quotient of the third argument by the first argument is calculated.
3. **Addition**:
   - 1 is added to all three arguments (`int1`, `int2`, and `float1`).
4. **Printing the Results**:
   - The results of the first set of calculations are printed on one line, separated by spaces.
   - Additional calculations are performed:
     - The first argument bitwise right shift by 3.
     - The second argument divided by 2 (using floating-point division).
     - The bitwise OR of the first and second arguments.
   - These results are printed on a new line, separated by spaces.
5. **Final Calculation**:
   - The sum of the first argument (after addition) and the total number of arguments (excluding the program name) is calculated and printed on the last line.