```python
#Sahil Patil BE3-38
import math

class BloomFilter:
    def __init__(self, items_count, bit_array_size):
        self.size = max(bit_array_size, 1)  # Ensure the bit array size is at least 1
        self.hash_count = max(1, self.get_hash_count(items_count, self.size))  # Ensure at least one hash function
        self.bit_array = [False] * self.size

    def add(self, item):
        for i in range(self.hash_count):
            digest = (hash(item) + i) % self.size
            self.bit_array[digest] = True

    def check(self, item):
        for i in range(self.hash_count):
            digest = (hash(item) + i) % self.size
            if not self.bit_array[digest]:
                return False
        return True

    @classmethod
    def get_hash_count(cls, n, m):
        '''Calculate number of hash functions based on the number of items and size of bit array'''
        if n == 0 or m == 0:
            return 0
        k = (m / n) * math.log(2)
        return int(k)

if __name__ == "__main__":
    # Parameters for Bloom filter
    n = int(input("Enter the expected number of items to add: "))
    bit_array_size = int(input("Enter the size of the bit array: "))

    # Create Bloom filter
    bloomf = BloomFilter(n, bit_array_size)
    print("Size of bit array: {}".format(bloomf.size))
    print("Number of hash functions: {}".format(bloomf.hash_count))

    # Adding numbers to Bloom filter
    print("Enter numbers to add to the Bloom filter (type 'done' to finish):")
    while True:
        input_value = input()
        if input_value.lower() == 'done':
            break
        try:
            number = int(input_value)
            bloomf.add(number)
            print(f"Added {number} to the Bloom filter.")
        except ValueError:
            print("Please enter a valid number.")

    # Check presence of numbers
    print("Enter numbers to check in the Bloom filter (type 'done' to finish):")
    while True:
        input_value = input()
        if input_value.lower() == 'done':
            break
        try:
            number = int(input_value)
            if bloomf.check(number):
                print(f"'{number}' may be present (possible positive).")
            else:
                print(f"'{number}' is definitely not present.")
        except ValueError:
            print("Please enter a valid number.")
```

```
Enter the expected number of items to add: 2
Enter the size of the bit array: 5
Size of bit array: 5
Number of hash functions: 1
Enter numbers to add to the Bloom filter (type 'done' to finish):
10
Added 10 to the Bloom filter.
7
Added 7 to the Bloom filter.
done
Enter numbers to check in the Bloom filter (type 'done' to finish):
14
'14' is definitely not present.
```

```
15
'15' may be present (possible positive).
done
```