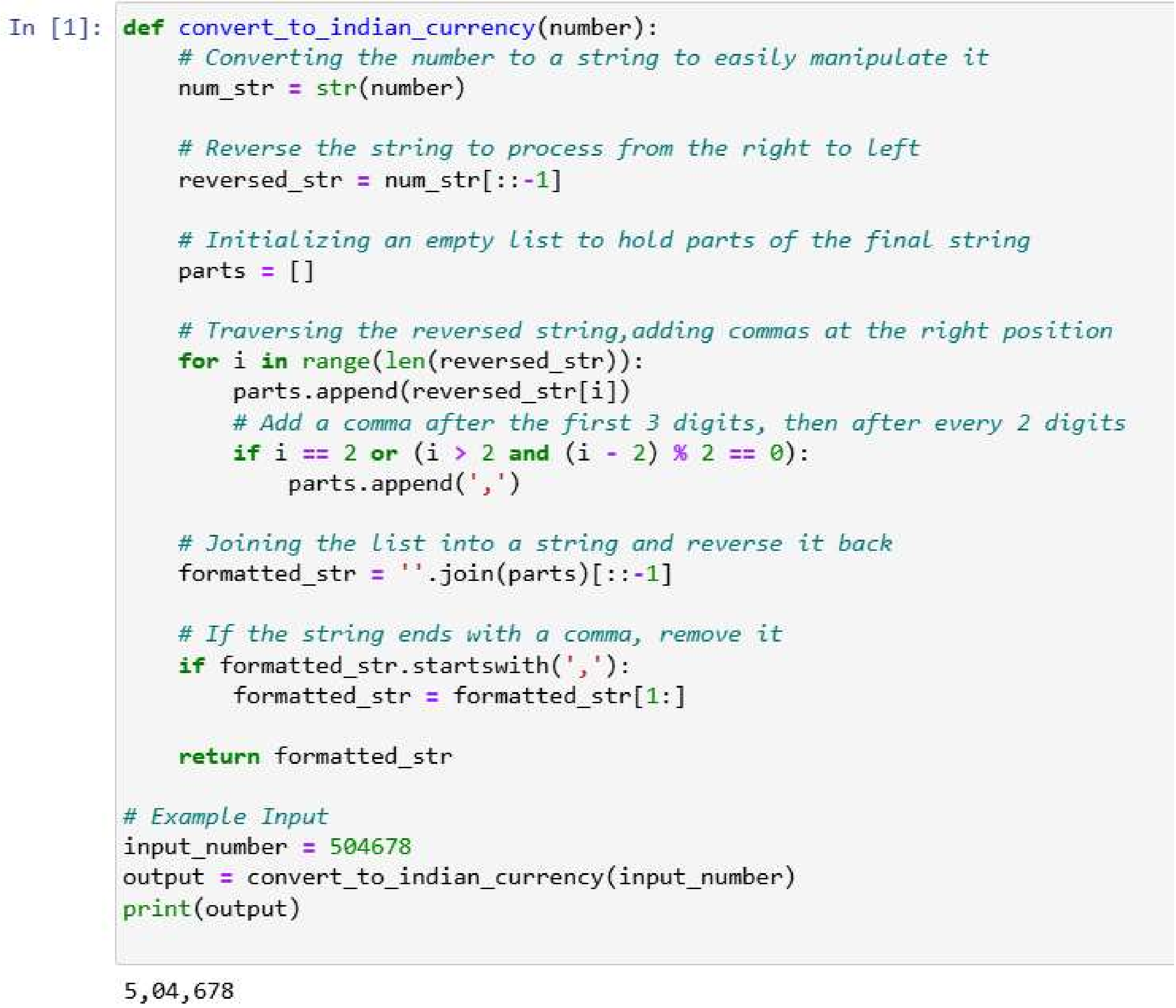
NAME: PARTH PRAVEEN SHETTY

Question 1

Write a python code for converting integer values to Indian currency notations, without using the currency libraries
Example:
input: 504678
output: 5,04,67

Answer:



In [1]:

```python
def convert_to_indian_currency(number):
    # Converting the number to a string to easily manipulate it
    num_str = str(number)

    # Reverse the string to process from the right to left
    reversed_str = num_str[::-1]

    # Initializing an empty list to hold parts of the final string
    parts = []

    # Traversing the reversed string,adding commas at the right position
    for i in range(len(reversed_str)):
        parts.append(reversed_str[i])
        # Add a comma after the first 3 digits, then after every 2 digits
        if i == 2 or (i > 2 and (i - 2) % 2 == 0):
            parts.append(',')

    # Joining the list into a string and reverse it back
    formatted_str = ''.join(parts)[::-1]

    # If the string ends with a comma, remove it
    if formatted_str.startswith(','):
        formatted_str = formatted_str[1:]

    return formatted_str

# Example Input
input_number = 504678
output = convert_to_indian_currency(input_number)
print(output)
```

```
5,04,678
```

Question 2

You won’t get caught if you hide behind someone.”

Sang-Woo advises Gi-Hun to hide behind someone to avoid getting shot.

Gi-Hun follows Sang-Woo's advice and hides behind Ali, who saved his life earlier. Gi-Hun and Ali both have the same height, K

. Many players saw this trick and also started hiding behind Ali.

Now, there are N

players standing *between* Gi-Hun and Ali in a straight line, with the ith player having height Hi

. Gi-Hun wants to know the minimum number of players who need to get shot so that Ali is visible in his line of sight.

Note:

- Line of sight is a straight line drawn between the topmost point of two objects. Ali is visible to Gi-Hun if nobody between them crosses this line.
- Even if there are some players who have the same height as that of Gi-Hun and Ali, Ali will be visible in Gi-Hun's line of sight.
- Gi-Hun and Ali have the same height.

Input Format

- The first line of input contains a single integer T

, denoting the number of test cases. The description of T

- test cases follows.
- The first line of each test case contains two space-separated integers N and K • , denoting the total number of players between Gi-Hun and Ali and the height of both of them respectively.
- The second line of each test case contains N

- space-separated integers, denoting the heights of the players between Gi-Hun and Ali.

Output Format

For each test case, output in a single line the minimum number of players who need to get shot so that Ali is visible in Gi-Hun's line of sight.

Constraints

- $1 \le T \le 105$
- $1 \le N \le 105$
- $1 \le K \le 106$
- $1 \le Hi \le 106$ for every $1 \le i \le N$

- .
- The sum of *N* across all test cases does not exceed 5·105
- .

Sample Input 1

```
3
4 10
2 13 4 16
5 8
9 3 8 8 4
4 6
1 2 3 4
```

Sample Output 1

```
2
1
0
```

Explanation

Test Case 1: Gi-Hun and Ali have height 10

. For Ali to be visible to Gi-Hun, the second person (with height 13) and the fourth person (with height 16) need to get shot. Hence, the minimum number of players who need to get shot is 2

.

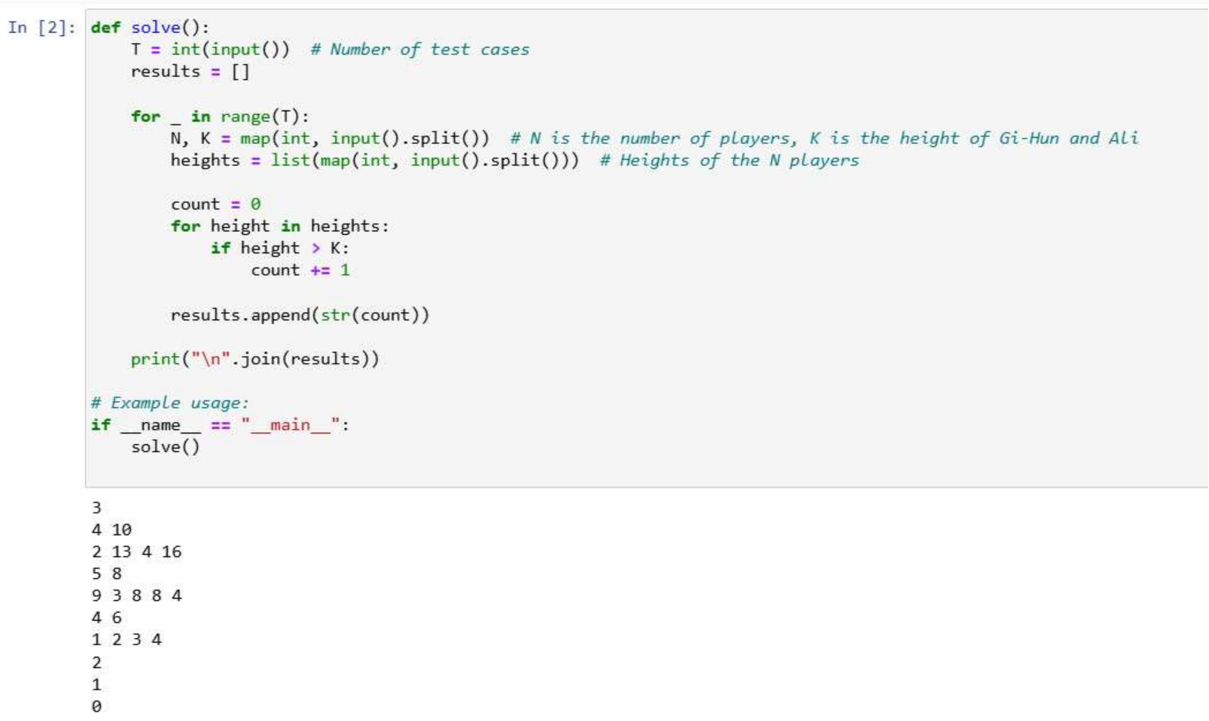
Test Case 2: Gi-Hun and Ali have height 8

. For Ali to be visible to Gi-Hun, the first person (with height 9) needs to get shot. Hence, the minimum number of players who need to get shot is 1

.

Test Case 3: Nobody needs to get shot because everyone is shorter than Gi-Hun and Ali.

Answer:



```
In [2]: def solve():
            T = int(input())  # Number of test cases
            results = []

            for _ in range(T):
                N, K = map(int, input().split())  # N is the number of players, K is the height of Gi-Hun and Ali
                heights = list(map(int, input().split()))  # Heights of the N players

                count = 0
                for height in heights:
                    if height > K:
                        count += 1

                results.append(str(count))

            print("\n".join(results))

        # Example usage:
        if __name__ == "__main__":
            solve()
```

```
3
4 10
2 13 4 16
5 8
9 3 8 8 4
4 6
1 2 3 4
2
1
0
```

CODE FOR ANSWERS

ANSWER 1:

```
def convert_to_indian_currency(number):
    # Converting the number to a string to easily manipulate it
    num_str = str(number)

    # Reverse the string to process from the right to left
    reversed_str = num_str[::-1]

    # Initializing an empty list to hold parts of the final string
    parts = []

    # Traversing the reversed string,adding commas at the right position
    for i in range(len(reversed_str)):
        parts.append(reversed_str[i])
        # Add a comma after the first 3 digits, then after every 2 digits
        if i == 2 or (i > 2 and (i - 2) % 2 == 0):
            parts.append(',')
    # Joining the list into a string and reverse it back
    formatted_str = ''.join(parts)[::-1]

    # If the string ends with a comma, remove it
    if formatted_str.startswith(','):
        formatted_str = formatted_str[1:]

    return formatted_str

# Example Input
input_number = 504678
output = convert_to_indian_currency(input_number)
print(output)
```

ANSWER 2:

```
def solve():
    T = int(input())  # Number of test cases
    results = []

    for _ in range(T):
        N, K = map(int, input().split())  # N is the number of players, K is the height of Gi-Hun and Ali
        heights = list(map(int, input().split()))  # Heights of the N players

        count = 0
        for height in heights:
            if height > K:
                count += 1
        results.append(str(count))

    print("\n".join(results))
# Example usage:
if __name__ == "__main__":
    solve()
```