```
def Hash(value, N):
    return value % N

def linear_prob(H, index, N):
    i = 1
    comparison = 0
    while H[index] != 0:
        comparison += 1
        n_index = (index + i) % N
        if H[n_index] == 0:
            return n_index, comparison
        i += 1
    return index, comparison

def quad_prob(H, index, N):
    i = 1
    comparison = 0
    while H[index] != 0:
        comparison += 1
        n_index = (index + i * i) % N
        if H[n_index] == 0:
            return n_index, comparison
        i += 1
    return index, comparison

N = int(input("Enter the number of clients: "))
H_linear = [0] * N
H_quad = [0] * N

linear_comparison = 0
quad_comparison = 0

for i in range(N):
    x = int(input(f"Enter the telephone number of client {i+1}: "))
    index = Hash(x, N)


    if H_linear[index] == 0:
        H_linear[index] = x
        print("Linear probing:", H_linear)
        linear_comparison += 1
    else:
        n_index, comparisons = linear_prob(H_linear, index, N)
        H_linear[n_index] = x
        linear_comparison += comparisons
        print("Linear probing:", H_linear)


    if H_quad[index] == 0:
        H_quad[index] = x
```

```
        print("Quadratic probing:", H_quad)
        quad_comparison += 1
    else:
        n_index, comparisons = quad_prob(H_quad, index, N)
        H_quad[n_index] = x
        quad_comparison += comparisons
        print("Quadratic probing:", H_quad)

print(f"\nTotal comparisons using linear probing: {linear_comparison}")
print(f"Total comparisons using quadratic probing: {quad_comparison}")
```