```
def initializeHashTable():
    size = int(input('Enter size of hash table: '))
    hashtable = [[-1, 'null'] for _ in range(size)]
    print('Hashtable of size', size, 'is successfully created.')
    return size, hashtable

choice = 0
while choice != 3:
    print('\n\nMenu')
    print('1. Linear Probing')
    print('2. Double Hashing')
    print('3. Exit')
    choice = int(input('Enter your choice: '))

    if choice == 1:
        size, hashtable = initializeHashTable()
        count = 0
        while True:
            print('\nMenu for Linear Probing')
            print('1. Insert')
            print('2. Search')
            print('3. Display')
            print('4. Back')
            choice1 = int(input('Enter your choice: '))

            if choice1 == 1:
                if count == size:
                    print('Hash table is Full.')
                else:
                    number = int(input('Enter number: '))
                    name = input('Enter Name: ')
                    hashvalue = number % size
                    original_hash = hashvalue
                    while hashtable[hashvalue][0] != -1:
                        print('Collision occurred. Using Linear Probing...')
                        hashvalue = (hashvalue + 1) % size
                        if hashvalue == original_hash:
                            print('Table is full. Insertion failed.')
                            break
                    else:
                        hashtable[hashvalue] = [number, name]
                        count += 1
                        print(f'Data inserted. Total records: {count}')

            elif choice1 == 2:
                number = int(input('Enter number to search: '))
                hashvalue = number % size
                comparisons = 0
                while hashtable[hashvalue][0] != number and comparisons < size:
                    hashvalue = (hashvalue + 1) % size
                    comparisons += 1
                if hashtable[hashvalue][0] == number:
                    print(f'Number {number} found at {hashvalue} with {comparisons + 1} compar
                else:
                    print(f'Number NOT found. Comparisons: {comparisons + 1}')
```

```
        elif choice1 == 3:
            for i in range(size):
                print(f'Hash {i} -> {hashtable[i]}')

        elif choice1 == 4:
            break

elif choice == 2:
    size, hashtable = initializeHashTable()
    prime = int(input('Enter a prime number less than size for Double Hashing: '))
    count = 0
    while True:
        print('\nMenu for Double Hashing')
        print('1. Insert')
        print('2. Search')
        print('3. Display')
        print('4. Back')
        choice1 = int(input('Enter your choice: '))

        if choice1 == 1:
            if count == size:
                print('Hash table is Full.')
            else:
                number = int(input('Enter number: '))
                name = input('Enter Name: ')
                hash1 = number % size
                hash2 = prime - (number % prime)
                i = 0
                inserted = False
                while i < size:
                    new_hash = (hash1 + i * hash2) % size
                    if hashtable[new_hash][0] == -1:
                        hashtable[new_hash] = [number, name]
                        count += 1
                        print(f'Data inserted at index {new_hash}. Total records: {count}'
                        inserted = True
                        break
                    else:
                        print('Collision occurred. Using Double Hashing...')
                    i += 1
                if not inserted:
                    print('Table is full or insertion failed.')

        elif choice1 == 2:
            number = int(input('Enter number to search: '))
            hash1 = number % size
            hash2 = prime - (number % prime)
            i = 0
            comparisons = 0
            found = False
            while i < size:
                new_hash = (hash1 + i * hash2) % size
                comparisons += 1
                if hashtable[new_hash][0] == number:
```

```
                    print(f'Number {number} found at {new_hash} with {comparisons} compari
                    found = True
                    break
                elif hashtable[new_hash][0] == -1:
                    break
                i += 1
            if not found:
                print(f'Number NOT found. Comparisons: {comparisons}')

        elif choice1 == 3:
            for i in range(size):
                print(f'Hash {i} -> {hashtable[i]}')

        elif choice1 == 4:
            break

print('Program exited.')
```

Program Output (Example)

```
Example Output:

Enter size of hash table: 5
Hashtable of size 5 is successfully created.
Menu for Linear Probing
1. Insert
2. Search
3. Display
4. Back
Enter your choice: 1
Enter number: 10
Enter Name: John
Data inserted. Total records: 1
Enter your choice: 3
Hash 0 -> [-1, 'null']
Hash 1 -> [-1, 'null']
Hash 2 -> [10, 'John']
Hash 3 -> [-1, 'null']
Hash 4 -> [-1, 'null']
Enter your choice: 4
Program exited.
```