Question 1 (25 pts):

Suppose there are eight students with IDs 236500360, 250987540, 262490004, 274052200, 283450902, 305342111, 317800977, and 328966568. Suppose hash table, HT, is of the size 13, indexed 0,1,2, . . ., 12. Show how these students' IDs, in the order given, are inserted in HT using the hashing function h(k) = k % 13, where k is a student ID.

IDs	HT index
236500360	5
250987540	11
262490004	10
274052200	6
283450902	7
305342111	9
317800977	0
328966568	8

Question 2 (25 pts)**:**

Suppose there are eight students with IDs 316500361, 330987540, 342490004, 344052199, 363450900, 385342111, 397800977, and 408966574. Suppose hash table, HT, is of the size 19, indexed 0, 1, 2, . . ., 18. Show how these students' IDs, in the order given, are inserted in HT using the hashing function h(k)= k % 19. Use linear probing to resolve collision.

IDs	HT index
316500361	14
330987540	16
342490004	13
344052199	9
363450900	15
385342111	17
397800977	10
408966574	11

Question 3 (25 pts)**:**

Suppose there are eight students with IDs 313577331, 330987544, 349722100, 354885711, 363450906, 375342113, 387800977, and 408966574. Suppose hash table, HT, is of the size 19, indexed 0, 1, 2, . . ., 18. Show how these students' IDs, in the order given, are inserted in HT using the hashing function h(k)= k % 19. Use quadratic probing to resolve collision.

IDs	HT index
313577331	1
330987544	2
349722100	6
354885711	7
363450906	5
375342113	10
387800977	14
408966574	11

Question 4:

By using the hashT class provided (hashT.h), which uses quadratic probing to resolve collision, create a hash table to keep track of student IDs and names.

(You can set your own hash table size)

Write a C++ code to;

(a) Ask the user to enter IDs and names of five students to be added to the hash table.
(b) Calculate the hash index from the student ID using the folding method.
(c) Add the information to the hash table.
(d) Ask the user to enter the student information to be deleted from the hash table.
(e) Remove the information from the hash table.
(f) Print the information.

Submit all the .cpp and .h files, and copy and paste the screenshot of the output here.

Sample output:

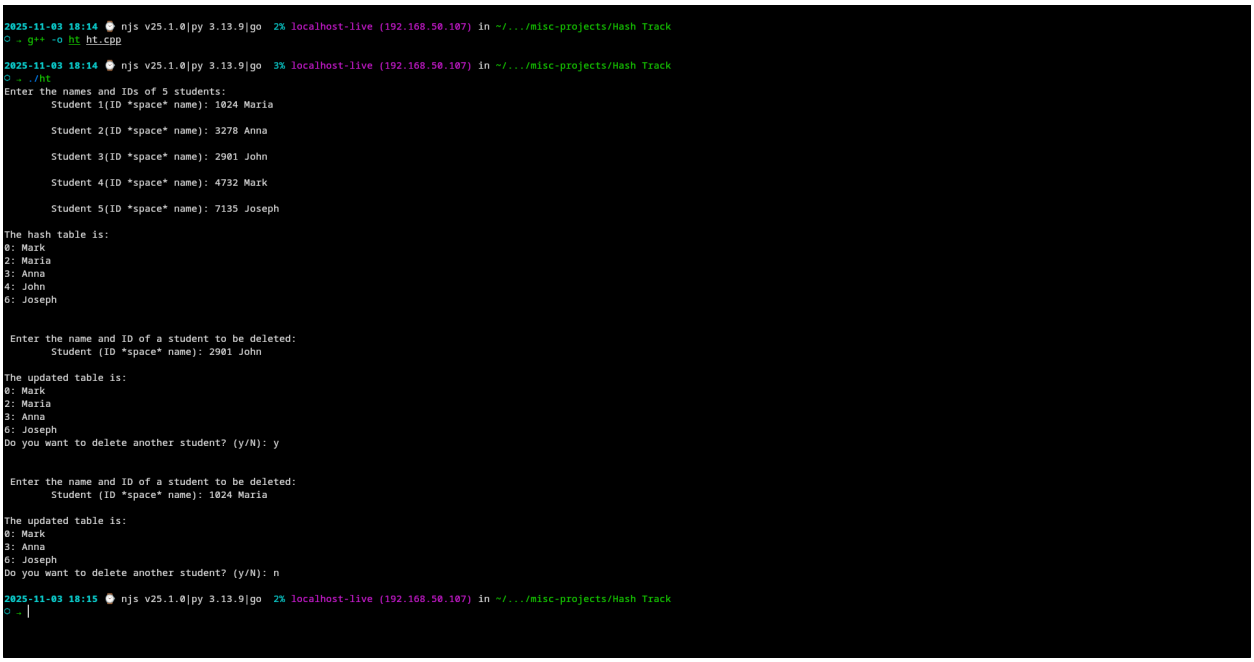
```
Enter the number of students:6
Enter ID's and names for 6 students:
1024 Maria
3278 Anna
2981 John
4732 Mark
7135 Joseph
9125 Bob

Students information in the harsh table
5: Joseph
9: Anna
10: John
15: Bob
34: Maria
79: Mark
Enter the ID and name of student information to delete: 2981 John

Students information in the harsh table
5: Joseph
9: Anna
15: Bob
34: Maria
79: Mark


...Program finished with exit code 0
Press ENTER to exit console.
```

My Output:


2025-11-03 18:14 njs v25.1.0|py 3.13.9|go 2% localhost-live (192.168.50.107) in ~/.../misc-projects/Hash Track
g++ -o ht ht.cpp
2025-11-03 18:14 njs v25.1.0|py 3.13.9|go 3% localhost-live (192.168.50.107) in ~/.../misc-projects/Hash Track
./ht
Enter the names and IDs of 5 students:
Student 1(ID *space* name): 1024 Maria
Student 2(ID *space* name): 3278 Anna
Student 3(ID *space* name): 2901 John
Student 4(ID *space* name): 4732 Mark
Student 5(ID *space* name): 7135 Joseph
The hash table is:
0: Mark
2: Maria
3: Anna
4: John
6: Joseph
Enter the name and ID of a student to be deleted:
Student (ID *space* name): 2901 John
The updated table is:
0: Mark
2: Maria
3: Anna
6: Joseph
Do you want to delete another student? (y/N): y
Enter the name and ID of a student to be deleted:
Student (ID *space* name): 1024 Maria
The updated table is:
0: Mark
3: Anna
6: Joseph
Do you want to delete another student? (y/N): n
2025-11-03 18:15 njs v25.1.0|py 3.13.9|go 2% localhost-live (192.168.50.107) in ~/.../misc-projects/Hash Track