Assignment 2

CSE341 File Organization
Fall 2023

Your task is to write member functions of a class that stores numbers using Computed Chaining Insertion. A header file with the class definition inside; a source file, and a main file are provided to you. In the source file(ComputedChaining.cpp), I defined some useful functions beforehand.

You have three functions to implement:

- **isThisYourHome:** The parameter passed to the function represents the key which is inserted previously into the hash table. It should return a boolean value whether the key is in the address of its hash function. You should note that this function should return true when there is no collision. **(10 pts)**

- **insert:** The parameter passed is the value to be inserted into the hash table. Write the necessary code to insert this value into the hash table using Computed Chaining. You should utilize from functions given to calculate the index you will insert into.
    - For the first part, you should consider adding the key to the chain when the collision occurs. **(25 pts)**
    - Secondly, you will implement the case when you move the key which is not at its home address. To simplify the question, some comments are added to ComputedChaining.cpp. You can follow the steps defined there. **(35 pts)**

- **find_num_probes:** This function's job is to find out how many calculations you make to get the key. Hint: The index of the key in the chain is related to the probe. **(30 pts)**

DO NOT make any changes to the header file and given code.
DO NOT send a main function along with your work.
DO NOT change the names of the files.

Note: Submit your own code. Similar codes will not be graded.

When you code the algorithm correctly, you will get the following result with the given input.

```
[redacted] g++ .\main.cpp .\ComputedChaining.cpp -o ComputedChaining.exe
[redacted] .\ComputedChaining.exe

---Table---
0:  55  1       P(55)   2
1:  76  -1      P(76)   3
2:  65  4       P(65)   1
3:  30  -1      P(30)   2
4:  11  -1      P(11)   1
5:  19  -1      P(19)   1
6:  90  3       P(90)   1
```