# CS2413: Assignment 4

Total Points: 100

Due: Nov 14, midnight.

## Primary Objectives

Be able to implement a direct address table whose size is dynamic, not knowing the range of input keys.

Be able to implement a hash table based on mod hash function whose modulus is dynamically input.

Be able to implement two collision resolutions: separate chaining and open addressing with quadratic probing.

## Project Overview

Our program should take a set of keys as input and use them to construct a hash table with any collision properly dealt with. Then, it should take a search key as input and use it to search for the satellite data.

## Project Specification

Our program should take four sets of inputs:

(i) an arbitrary sequence of non-repeated keys and their associated satellite data. Each key is a positive integer and each satellite data is character of one letter such as 'A' or 'f'.

Take your input by the following format:[1]

- first, input a sequence of keys and end it with -1.[2]

- then, input a sequence of satellite data and end it with '*'.[3]

(ii) an integer indicating what table is being implemented. '0' means direct address table, and '1' means hash table with collisions addressed by separate chaining, and '2' means hash table with collisions addressed by open addressing plus quadratic probing.

(iii) an integer as the modulus of the hash function.

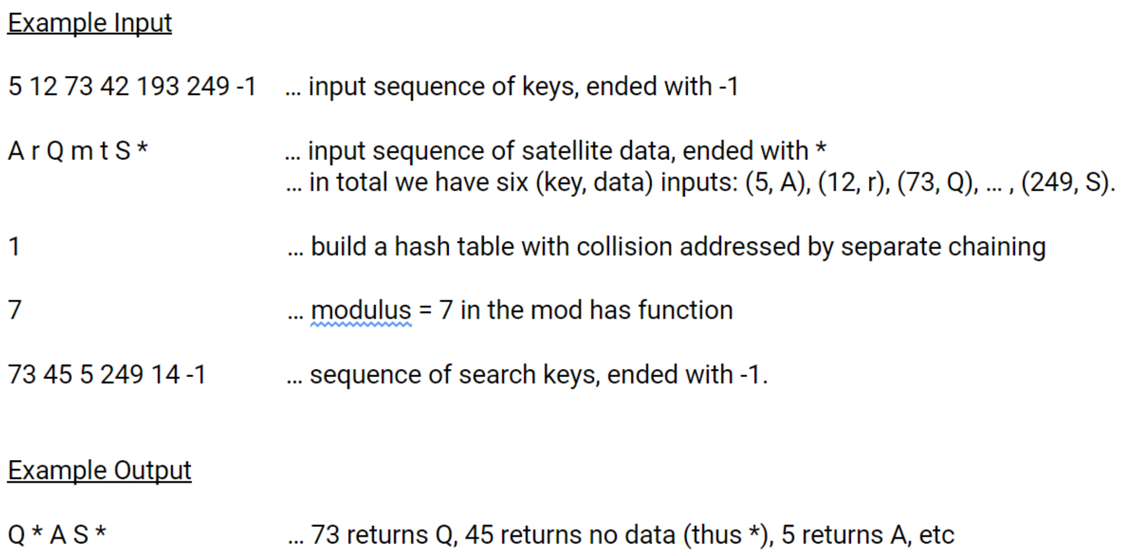
(iv) a sequence of search keys ended with -1, used to search for satellite data when the table is constructed.

Our program should give one set of outputs:

(i) the set of satellite data retrieved from the table based on the search keys. If a search key is not found, return satellite data '*'.

An example input and output are in Figure 1.

[1] We will see an example input and output in Figure 1.
[2] Note that keys can only be positive integers so '-1' shall not be considered as a valid key.
[3] Note that satellite data can only be letters so '*' shall not be considered as a valid satellite data.



Example Input

```
5 12 73 42 193 249 -1   … input sequence of keys, ended with -1

A r Q m t S *           … input sequence of satellite data, ended with *
                        … in total we have six (key, data) inputs: (5, A), (12, r), (73, Q), … , (249, S).

1                       … build a hash table with collision addressed by separate chaining

7                       … modulus = 7 in the mod has function

73 45 5 249 14 -1       … sequence of search keys, ended with -1.
```

Example Output

```
Q * A S *               … 73 returns Q, 45 returns no data (thus *), 5 returns A, etc
```

**Fig. 1.**

## Additional Requirements and Tips

When addressing collision using open addressing with quadratic probing, you need to make sure all keys are placed in the hash table, meaning that if all probed slots in the existing table are occupied, you need to increase the table size to accommodate them.

Choose proper data structures yourself.

Note that, for direct address table, you do not know the range of input keys upfront. Similarly, for open addressing scheme, you do not know whether all probed slots will be occupied or not upfront.

You can use the 'vector' library, but not others that offer functions directly related to hash table.

To enable Gradescope grading, please only use 'cin' and 'cout' for data input and output.

Please name your submitted code as `cs2413_hw4.cpp`.

* We try best to avoid vagueness in the designed tasks. But if you notice any, please contact the instructor and we will fix it and update it on Canvas if necessary.

## Rubrics

- direct address table: 25 points.

- hash table with separate chaining: 30 points.

- hash table with open addressing (quadratic probing): 35 points.

- Documentation: 10 points.