

Airport Code Lookup

Closed Hashing Dictionary

Your goal is to simulate a closed hashing dictionary using the **Cursor Based** implementation and **Progressive Overflow**. The given set, hash logic, and virtual heap illustration are provided below. Your final output should be a document that includes (1) the final state of the virtual heap; and (2) the hash index for each code in the set.

Set A = {"JFK", "LAX", "SFO", "CDG", "LHR", "NRT", "ATL", "DXB", "PEK", "MIA", "SYD", "ORD"}

hash(code) == ((code[0] - 'A') * 26 * 26 + (code[1] - 'A') * 26 + (code[2] - 'A')) % 10

Example:
hash("JFK") -> 4

INITIAL

elem		next	elem		next
0		-1	10		11
1		-1	11		12
2		-1	12		13
3		-1	13		14
4		-1	14		15
5		-1	15		16
6		-1	16		17
7		-1	17		18
8		-1	18		19
9		-1	19		-1

primary

secondary

Avail

10

FINAL STATE

elem		next	elem		next
0	MIA	-1	10	NRT	^{11, -1} 14
1		-1	11	ATL	^{12, -1} 13
2	SFO	-1	12	PEK	-1
3		-1	13	SYD	-1
4	JFK	⁻¹ 12	14	ORD	-1
5	LHR	⁻¹ 11	15		16
6	CDG	-1	16		17
7	DXB	-1	17		18
8		-1	18		19
9	LAX	⁻¹ 10	19		-1

primary

secondary

Avail

15

10

11

12

13

14

1. JFK
 $74 - 65 * 26 * 26 \quad 6084 + 130 + 10$
 $70 - 65 * 26$
 $75 - 65 \quad 6224 \% 10 = 4$

2. LAX
 $76 - 65 * 26 * 26 \quad 7436 + 0 + 23$
 $65 - 65 * 26$
 $88 - 65 \quad 7459 \% 10 = 9$

3. SFO
 $83 - 65 * 26 * 26 \quad 12168 + 130 + 14$
 $70 - 65 * 26$
 $79 - 65 \quad 12312 \% 10 = 2$

4. CDG
 $67 - 65 * 26 * 26 \quad 1352 + 78 + 6$
 $68 - 65 * 26$
 $71 - 65 \quad 1436 \% 10 = 6$

5. LHR
 $76 - 65 * 26 * 26 \quad 7436 + 182 + 17$
 $72 - 65 * 26$
 $82 - 65 \quad 7635 \% 10 = 5$

6. NRT
 $78 - 65 * 26 * 26 \quad 8788 + 442 + 19$
 $82 - 65 * 26$
 $84 - 65 \quad 9249 \% 10 = 9$

COLLISION WITH LAX

7. ATL
 $65 - 65 * 26 * 26 \quad 0 + 494 + 11$
 $84 - 65 * 26$
 $76 - 65 \quad 505 \% 10 = 5$

COLLISION WITH LHR

8. DXB
 $68 - 65 * 26 * 26 \quad 2028 + 598 + 1$
 $88 - 65 * 26$
 $66 - 65 \quad 2627 \% 10 = 7$

9. PEK
 $80 - 65 * 26 * 26 \quad 10140 + 104 + 10$
 $69 - 65 * 26$
 $75 - 65 \quad 10254 \% 10 = 4$

COLLISION WITH JFK

10. MIA
 $77 - 65 * 26 * 26 \quad 8112 + 208 + 0$
 $73 - 65 * 26$
 $65 - 65 \quad 8320 \% 10 = 0$

11. SYD
 $83 - 65 * 26 * 26 \quad 12168 + 624 + 3$
 $89 - 65 * 26$
 $68 - 65 \quad 12795 \% 10 = 5$

COLLISION WITH LHR, ATL

12. ORD
 $79 - 65 * 26 * 26 \quad 9464 + 442 + 3$
 $82 - 65 * 26$
 $68 - 65 \quad 9909 \% 10 = 9$

COLLISION WITH LAX, NRT

code for hash idx ONLY (kinda lazy to find ASCII value for each and compute it manually...)

```
C/C++
#include <string.h>
#include <ctype.h>
#include <stdio.h>

int funcHash(char *set) {
    printf("%c:%d -- %d\n", set[0], set[0], (set[0]-'A') * 26 * 26);
    printf("%c:%d -- %d\n", set[1], set[1], (set[1]-'A') * 26);
    printf("%c:%d -- %d\n", set[2], set[2], (set[2]-'A'));

    printf("sum: %d\n", (set[0]-'A') * 26 * 26 + (set[1]-'A') * 26 + (set[2]-'A'));

    return((set[0]-'A') * 26 * 26 + (set[1]-'A') * 26 + (set[2]-'A')) % 10;
}

int main() {
    char set[3]; // or 4 for allowance of '\0'

    printf("enter string: ");
    scanf(" %s", set);

    if(
        set[0] == '\0' ||
        set[1] == '\0' ||
        set[2] == '\0'
    ) return 0;

    // in case of lowercase input
    for(int i=0; i<3; i++) { set[i] = toupper(set[i]); }

    printf("hash: %d\n", funcHash(set));

    return 0;
}
```