

main.cpp

```
1 // Lina Kang
2 // CS1D MW 2:30 - 5:00 PM
3 // Assignment 7 - Hashing
4 // This program exercises the hashing algorithms using a map
5
6 /* OUTPUT
7 Lina Kang
8 CS1D MW 2:30-5:00PM
9 Assignment 7 - Hashing
10 This program exercises the hashing algorithms using a map
11 Data Structure Used: map
12 -----
13 Part One:
14 -----
15 (0, San Diego)
16 (1, El Segundo)
17 (2, Los Angeles)
18 (3, San Diego)
19 (4, San Clemente)
20 (6, Tustin)
21 (9, Vista)
22 (10, Dana Point)
23 (11, La Jolla)
24 (12, San Juan)
25 (13, Santa Ana)
26 (14, Laguna Beach)
27 (15, Irvine)
28 (16, Oceanside)
29 (17, Orange)
30 (18, Laguna Niguel)
31 (19, Brea)
32 (20, Anaheim)
33 (22, San Clemente)
34 (23, Aliso Viejo)
35 (26, Del Mar)
36 (28, Laguna Hills)
37 -----
38 Part Two:
39 -----
40 (0, Los Angeles)
41 (1, San Diego)
42 (3, San Diego)
43 (4, Tustin)
44 (5, Laguna Hills)
45 (6, San Juan)
46 (10, San Clemente)
47 (11, Riverside)
48 (12, La Jolla)
49 (13, Irvine)
50 (16, Laguna Beach)
51 (17, Orange)
52 (18, Laguna Niguel)
53 (19, Anaheim)
54 (20, Brea)
55 (22, San Clemente)
56 (23, Del Mar)
57 (25, Vista)
```

```

58 (26,El Segundo)
59 (27,Aliso Viejo)
60 (29,Santa Ana)
61 (30,Dana Point)
62 */
63
64 #include <iostream>
65 #include <fstream>
66 #include <string>
67 #include <map>
68
69 using namespace std;
70
71 //main function that does all the hashing
72 void hashing(int (*hashFunc)(int),
73             void (*insert)(int, string, map<int, string> &));
74
75 // Part 1. hashing helper functions
76 int hashOne(int key);
77 int hashDouble(int key);
78 void insertKeyDouble(int key, string line, map<int, string> &hashmap);
79
80 // Part 2. hashing helper functions
81 int hashTwo(int key);
82 void insertKeyQuadratic(int key, string line, map<int, string> &hashmap);
83
84 int main()
85 {
86     cout << "Lina Kang\n"
87            "CS1D MW 2:30-5:00PM\n"
88            "Assignment 7 - Hashing\n"
89            "This program exercises the hashing algorithms using a map\n";
90     cout << "Data Structure Used: map  \n"
91            "-----\n"
92            "Part One: \n"
93            "-----\n";
94
95     hashing(hashOne,insertKeyDouble);
96
97     cout << "-----\n"
98            "Part Two: \n"
99            "-----\n";
100
101     hashing(hashTwo,insertKeyQuadratic);
102
103 }
104
105 // main function that does all the hashing
106 void hashing(int (*hashFunc)(int), void (*insert)(int, string, map<int, string> &))
107 {
108     map<int, string> hashmap;
109     map<int, string>::iterator p;
110
111     ifstream input ("input.txt");    // read in input from a separate txt file
112
113     string line;
114     int key;

```

```

115
116 while(input >> line)
117 {
118     if(line[0] == 'D') // if the line begins with "Delete" - erase key
119     {
120         input >> line;
121         input >> key;
122         key = hashFunc(key);
123         if(hashmap[key] != "")
124             hashmap.erase(key);
125     }
126     else // insert key and its info
127     {
128         key = stoi(line);
129         input.ignore(100, ' ');
130         getline(input, line);
131         insert(key, line, hashmap);
132     }
133 }
134
135 // print the map
136 for(p = hashmap.begin(); p != hashmap.end(); ++p)
137 {
138     cout << "(" << p->first << "," << p->second << ")\n";
139 }
140 }
141
142 // Part 1. hashing helper functions
143 int hashOne(int key)
144 {
145     return key % 29;
146 }
147 int hashDouble(int key)
148 {
149     return 13 - key % 13;
150 }
151 void insertKeyDouble(int key, string line, map<int, string> &hashmap)
152 {
153     int i = 0;
154     do
155     {
156         key = hashOne(hashOne(key) + i*hashDouble(key));
157         i++;
158     }while(hashmap[key] != "");
159     hashmap[key] = line;
160 }
161
162 // Part 2. hashing helper functions
163 int hashTwo(int key)
164 {
165     return key % 31;
166 }
167 void insertKeyQuadratic(int key, string line, map<int, string> &hashmap)
168 {
169     int i = 0;
170     do
171     {

```

main.cpp

```
172         key = hashTwo(hashTwo(key) + i*i);
173         i++;
174     }while(hashmap[key] != "");
175     hashmap[key] = line;
176 }
177
178 /* input.txt
179
180 18 Laguna Niguel
181 41 Mission Viejo
182 22 San Clemente
183 44 Irvine
184 Delete key 41
185 58 Lake Forest
186 32 San Diego
187 49 Anaheim
188 Delete key 58
189 31 Los Angeles
190 17 Orange
191 72 Palms Springs
192 41 Riverside
193 Delete key 72
194 19 Brea
195 60 Santa Ana
196 35 Tustin
197 103 Oceanside
198 11 La Jolla
199 18 Del Mar
200 22 Aliso Viejo
201 49 Laguna Beach
202 Delete key 41
203 42 Vista
204 49 San Diego
205 99 San Juan
206 29 Dana Point
207 88 El Segundo
208 41 San Clemente
209 62 Laguna Hills
210
211 */
212
```