#include <iostream>

#include <iomanip>

#include <vector>

#include <algorithm>

#include <functional>

int insert(int value, std::vector<int> &table) {

int Hash = value%1000;

int count = 0;

if(table[Hash]==-1){

table[Hash] = value;

}

else{

while (table[Hash+count]!=-1){

count++;

};

table[Hash+count] = value;

}

// Code to insert value into a hashed location in table

// where table is a vector of length 1000.

// Returns the number of collisions encountered when

// trying to insert value into table.

return count;

}

int main() {

// Prepare some random but distinct values

constexpr int NUM\_VALUES = 500;

std::vector<int> value(NUM\_VALUES);

int prev\_value = 0;

for (int i = 0; i < NUM\_VALUES; i++) {

prev\_value += rand()%25 + 1;

value[i] = prev\_value;

}

// Create hash table of size 1000 initialized with -1

std::vector<int> table(1000,-1);

// Insert values and track the maximum number of collisions

int max\_hit = 0, max\_value = -1;

for (int i = 0; i < NUM\_VALUES; i++) {

int hit = insert(value[i],table);

if (hit > max\_hit) {

max\_hit = hit;

max\_value = value[i];

}

}

std::cout << "Inserting value " << max\_value << " experienced " << max\_hit << " collisions." << std::endl <<std::endl;

// Print the table contents

for (int j = 0; j < 1000; j += 10) {

std::cout << std::setw(3) << j << ":";

for (int i = 0; i < 10; i++) {

if (table[j+i] == -1)

std::cout << " ";

else

std::cout << std::setw(6) << table[j+i];

}

std::cout << std::endl;

}

return 0;

}