Name: Siddhant Kumar Sahu

Batch: E3, 57

PRN: 202301070159

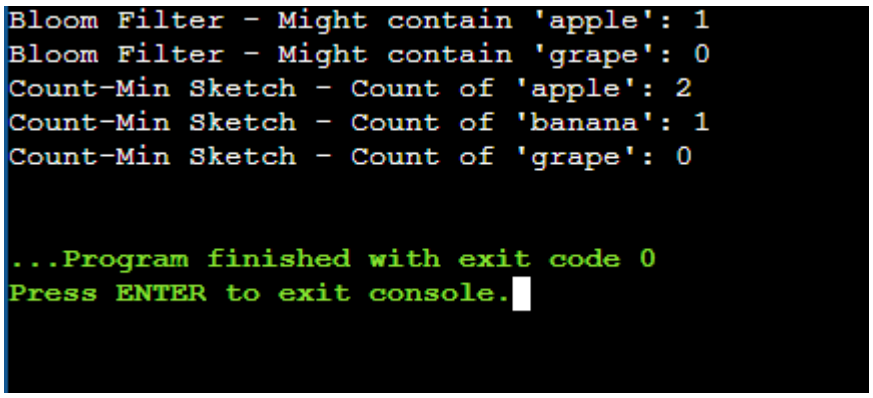
Practical - 6

CODE

```
#include <bits/stdc++.h>
using namespace std;

// ---------- Bloom Filter Implementation ----------
class BloomFilter {
    bitset<1000> bits;
    int size;
public:
    BloomFilter(int sz = 1000) : size(sz) {}

    void insert(const string& key) {
        hash<string> hashFn;
        size_t h1 = hashFn(key) % size;
        size_t h2 = (hashFn(key + "salt") * 7) % size;
        bits[h1] = 1;
        bits[h2] = 1;
    }

    bool mightContain(const string& key) {
        hash<string> hashFn;
        size_t h1 = hashFn(key) % size;
        size_t h2 = (hashFn(key + "salt") * 7) % size;
        return bits[h1] && bits[h2];
    }
};

// ---------- Count-Min Sketch Implementation ----------
class CountMinSketch {
    vector<vector<int>> table;
    int depth, width;
    vector<hash<string>> hashFuncs;

public:
    CountMinSketch(int d = 4, int w = 100) : depth(d), width(w) {
        table.resize(depth, vector<int>(width, 0));
```

```
    }

    void insert(const string& key) {
        for (int i = 0; i < depth; i++) {
            size_t hashVal = hash<string>{}(key + to_string(i));
            int index = hashVal % width;
            table[i][index]++;
        }
    }

    int estimateCount(const string& key) {
        int minCount = INT_MAX;
        for (int i = 0; i < depth; i++) {
            size_t hashVal = hash<string>{}(key + to_string(i));
            int index = hashVal % width;
            minCount = min(minCount, table[i][index]);
        }
        return minCount;
    }
};

// ---------- Main Function ----------
int main() {
    BloomFilter bloom;
    CountMinSketch cms;

    bloom.insert("apple");
    bloom.insert("banana");

    cms.insert("apple");
    cms.insert("apple");
    cms.insert("banana");

    cout << "Bloom Filter - Might contain 'apple': " <<
bloom.mightContain("apple") << endl;
    cout << "Bloom Filter - Might contain 'grape': " <<
bloom.mightContain("grape") << endl;

    cout << "Count-Min Sketch - Count of 'apple': " <<
cms.estimateCount("apple") << endl;
    cout << "Count-Min Sketch - Count of 'banana': " <<
cms.estimateCount("banana") << endl;
    cout << "Count-Min Sketch - Count of 'grape': " <<
cms.estimateCount("grape") << endl;

    return 0;
}
```

OUTPUT


Bloom Filter - Might contain 'apple': 1
Bloom Filter - Might contain 'grape': 0
Count-Min Sketch - Count of 'apple': 2
Count-Min Sketch - Count of 'banana': 1
Count-Min Sketch - Count of 'grape': 0
...Program finished with exit code 0
Press ENTER to exit console.