

## 1.4 DATA TYPES &amp; STRUCTURES · 1.4.2

## Hash tables

Original practice questions · 30 marks · about 40 minutes · spec 1.4.2(b)(c)

**Instructions.** Answer all questions. Marks are shown in brackets [ ]. Show your working for any hashing.

**1** Total: 4 marks

(a) State the purpose of a *hash function* in a hash table. [2]

---

---

(b) Explain why a hash table allows items to be found quickly. [2]

---

---

**2** Total: 8 marks

A hash table of size 10 uses the hash function  $\text{address} = \text{key} \text{ MOD } 10$ . The keys 41, 23, 67, 33 and 13 are inserted in this order. Collisions are resolved by placing the item in the next free cell (linear probing).

(a) Complete the table to show the address at which each key is stored. Show the hash for each. [5]

| Key        | 41 | 23 | 67 | 33 | 13 |
|------------|----|----|----|----|----|
| key MOD 10 |    |    |    |    |    |
| Stored at  |    |    |    |    |    |

(b) Identify **one** collision that occurred and explain how it was resolved. [2]

---

---

(c) Describe how the value with key 33 would be located when searching the table. [1]

---

---

3

Total: 6 marks

This question is about collisions and deletion.

(a) State what is meant by a *collision*. [1]

(b) Describe **one** method of resolving collisions other than placing the item in the next free cell. [2]

(c) A programmer deletes an item by simply emptying its cell. Explain why this can cause a later search to fail to find an item that is still in the table, and how this is prevented. [3]

4

Total: 5 marks

(a) State what is meant by the *load factor* of a hash table, and how it is calculated. [2]

(b) A table of size 200 holds 150 items. Calculate the load factor. [1]

(c) Explain the effect of a high load factor on the performance of the hash table. [2]

5

Total: 7 marks

A company stores its customer records and needs to look up a record by customer ID very quickly. A colleague suggests using a hash table; another suggests keeping the records in a sorted array searched with binary search.

**Discuss** whether a hash table is a suitable choice for this application, comparing it with the sorted-array alternative. Refer to lookup speed, collisions and the load factor in your answer. [7]

**END OF QUESTIONS**