# Homework 4 Hash tables Answers

1. The abstract data types (ADT) covered include lists, stacks, queues, hash tables, and dictionaries.

Complete the table to show the most suitable ADT in each case:

|  |  |
| --- | --- |
| **Application** | **ADT** |
| Storing a large data set that is going to be searched frequently | Hash table or dictionary |
| Controlling items to be printed | Queue |
| Storing data that is going to be processed from beginning to end | List/array/file |
| Holding parameters and return addresses from subroutines | Stack |

 [4]

2. A hash table of size 11 has been created using the hashing function:

**address = key mod (table size)**

 (a) Show in Table 1 at which addresses the values 37, 44, 92, 76, 87, 22 would be held. [3]

**Slots**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 44 | 87 | 22 |  | 37 | 92 |  |  |  |  | 76 |

*Table 1*

 (b) Explain how the items 37 and 22 would be located in the table. [2]

 The hashing algorithm is applied to the key 37, giving its address. The key 22 is hashed and the resulting address is 0, which contains a different key. Successive cells are searched until 22 is found.

 (c) A procedure **findItem(x)** to find an item returns **False** if the item cannot be found. Explain how the procedure will be able to ascertain that an item is not in the hash table. [2]

 If the address of an item, when hashed, contains a different value, successive cells are searched. If a blank cell is encountered, then the item is not in the table and **False** will be returned.

 (e) Explain what would need to be done to prevent the procedure **findItem(x)** from returning **False** when the item is in fact in the table. Assume some items have been deleted from the table. [2]

 When items are deleted from the table, a marker must be placed in the cell to differentiate it from an “empty” cell which has never been filled. The search for a particular item will then continue until an empty cell is found.

 (f) Explain, with the aid of an example, what is meant by the term “load factor” in connection with the hash table. Why is it advisable to avoid having a load factor close to 100%? [3]

 Load factor is the percentage of cells in the table which are filled. e.g. in a table of 100 items if 65 are filled, the load factor is 65%.

 A very high load factor will result in **many collisions**, which affect the **performance** of the hash table.

3. Internal telephone extension numbers in a large organisation are held in a dictionary data structure. Sample entries are:

{“Jones, A”: 352; “Arnot, G”: 101, “Harrison, M”:56, …}

 (i) What value will be returned by a lookup operation using the key “Arnot, G”? [1]

 101

 (ii) Explain why the dictionary entries are not held in alphabetical order. [2]

 The keys are hashed using a hashing algorithm to find their place in the dictionary data structure so will be in random sequence.

 (iii) Some employees and departments have more than one phone number to be stored in the dictionary. Show with the aid of an example how this can be achieved. [1]

 A list can be used to hod the values associated with a key.

 {“Jones, A”: [352,456]; “Arnot, G”: [101], “Harrison, M”:[56, 378,888]…}

 Total 20 marks