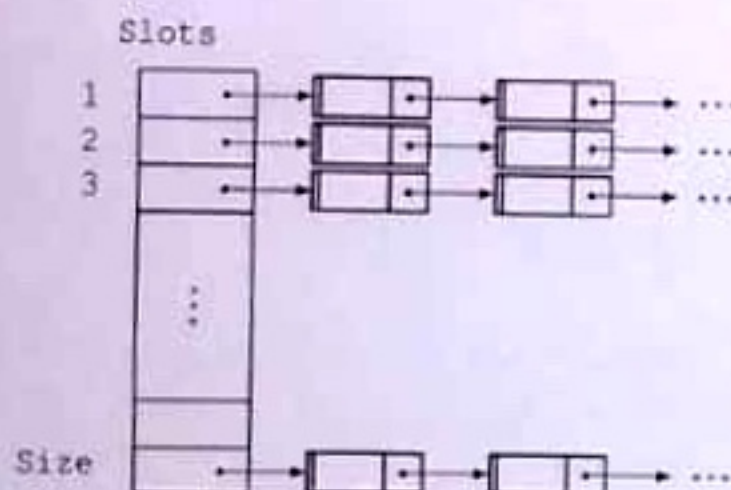
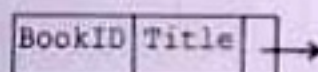


## PROGRAMMING CHALLENGE 8: HASHTABLE CHAINING

The task is to store records of books in a hash table and have direct access to records. Design the hash table as an array of linked list to handle collision of records. A hash function mapped into a location in the array, which contains a pointer to a linked list of records. When a collision occurs, records with the same hash code will be chained together.



Each node in the linked list contains a record of a book and implemented as an instance of the class `BookRec`.



The class `BookRec` has the following properties:

Class: <code>BookRec</code>	
Properties	
Identifier	Description
<code>BookID</code>	ID of a book
<code>Title</code>	Title of a book
<code>Pointer</code>	Pointer of node

### Task 1

Write program code for the class `BookRec`, using suitable data types for the attributes.

**Evidence 1:** Program code for Task 1.

[3]

A linked list is implemented as an instance of the class `LinkedList`. The class `LinkedList` has the following properties and methods:

Class: <code>LinkedList</code>	
Properties	
Identifier	Description
<code>Start</code>	Points to the first node of the linked list
Methods	
<code>Initialise</code>	Initialises <code>Start</code> to be Null.
<code>IsEmpty</code>	Tests for empty linked list. Returns Boolean.
<code>AddNode</code>	Adds a new node to the linked list. Takes <code>BookID</code> and <code>Title</code> as arguments.
<code>DeleteNode</code>	Deletes a node from the linked list. Takes <code>BookID</code> as argument.
<code>SearchNode</code>	Searches for an existence of a record. Takes <code>BookID</code> as argument.
<code>DisplayLinkedList</code>	Displays contents of linked list.

## Task 2

Write program code for the class `LinkedList`.

Evidence 2: Program code for Task 2.

[16]

The hash table is implemented as an instance of the class `HashTable`. The class `HashTable` has the following properties and methods:

Class: <code>HashTable</code>	
Properties	
Identifier	Description
<code>Size</code>	Size of the array
<code>Slots</code>	Array of linked list that makes up the hash table
Methods	
<code>Initialise</code>	• Initialises <code>Size</code> and <code>Slots</code> .
<code>Hash</code>	• A hashing function that calculates the address of the hash table
<code>Display</code>	• Displays the contents of the hash table.
<code>Put</code>	• Puts a record into the hash table after calculating the address to store the record. • Takes <code>BookID</code> and <code>Title</code> as arguments.
<code>Remove</code>	• Removes a record from the hash table. • Takes <code>BookID</code> as argument.
<code>Search</code>	• Searches for a record in the hash table. • Takes <code>BookID</code> as argument. • Returns Boolean.



The address in the hash table is calculated from a **hashing function** as follows:

- The ASCII code is calculated for each character within the `BookID`.
- The total of all ASCII values is calculated.
- The total is divided by `Size` and the remainder calculated with modulo arithmetic, where `Size` is the maximum address location.
- The value returned by the function is the  $(\text{remainder} + 1)$ . This value is the address for the record in the hash table.

For example, for a `Size` of value 17, if the book record has `BookID CS733`, the value from the hashing function is 2. Therefore, write record with `BookID CS733` to the array with address 2.

### Task 3

Write program code for the class `HashTable`, including the `Hash` function which uses the following specifications:

FUNCTION `Hash (BookID: STRING) RETURNS INTEGER`

Evidence 3: Program code for Task 3.

[16]

### Task 4

Test your program code for `Size` of value 17, by

- a. Putting the following records into the hash table and display the hash table.

<u>BookID</u>	<u>Title</u>
CS733	Basic algorithms
AB944	Master Computing
KS293	Data structures
BK232	Programming exercises
PK199	Testing Python

- b. Removing AB944 from the hash table and display the hash table.

- c. Searching for KS293 from the hash table.

Evidence 4: Screenshots of running Task 4.

[2]

[Total: 37 marks]