

APOLLO

B4 Abstract data types-HL only

HL

Computer Science

First assessment 2027

4 questions | mark schemes

4. (a) Award [3 max]

The *HashMap* is set up using pairs of string values;
representing the airport code and airport name / diagram to show this;
when the airport code is used as a parameter for the *HashMap*, the matching/corresponding
value of the airport name will be returned;

(b) Award [2 max]

the arrays are combined to form a single set;
the set will automatically eliminate any duplicates / will only hold unique values;

(c) (i) Award [1 max]

false;

(ii) Award [1 max]

[2,6];

6. (a) Award [5 max]*Award [1] for making use of a temporary node;**Award [1] for Boolean;**Award [1] for list traversal;**Award [1] for ending traversal if found;**Award [1] for correct comparison;**Award [1] for returning the correct values (in both cases);*

```

public void findPlayer(String name)
{
    Node temp = head;
    boolean found = false;
    while(temp != null)
    {
        if (temp.name.compareTo(name) == 0) // allow = name
        {
            System.out.println(temp.score);
            found = true;
            break;
        }
        else
            temp = temp.next;
    }
    if (found == false)
        System.out.println("Not Found");
}

```

(b) Award [5 max]

The algorithm will:

- make a temporary node equal to `head`;
- make use of a “previous” node;
- traverse the linked list (using the temporary node) comparing the score field of each node to the score field of the new node;
- until the insertion point is found;
- if insertion is at the beginning, link the new node to the head and make the head equal to the new node;
- if insertion is at the end, link the previous node to the new node;
- copy the *next* pointer of the node previous to the insertion point to the *next* pointer of the new node;
- link the previous node to the new node;

Note: Allow the use of diagrams to illustrate any of the above points.

Option B — Python

10. (a) Award [3 max]

The dictionary is set up using pairs of string values;
representing the airport code and airport name / diagram to show this;
when the airport code is used as a parameter for the dictionary, the matching/corresponding
value of the airport name will be returned;

(b) Award [2 max]

the lists are combined to form a single set/HashSet;
the set/HashSet will automatically eliminate any duplicates / will only hold unique values;

(c) (i) Award [1 max]

false;

(ii) Award [1 max]

{2, 6};

12. (a) Award [5 max]

Award [1] for making use of a temporary node;
Award [1] for Boolean;
Award [1] for list traversal;
Award [1] for ending traversal if found;
Award [1] for correct comparison;
Award [1] for returning the correct values (in both cases);

```
def find_player(self, name):
    temp = self.head
    found = False
    while temp != None:
        if temp.name == name:
            found = True
            print (temp.score)
            break
    else:
        temp = temp.next
    if found == False:
        print ("Not found")
```

(b) Award [5 max]

The algorithm will:

- make a temporary node equal to `head`;
- make use of a “previous” node;
- traverse the linked list (using the temporary node) comparing the score field of each node to the score field of the new node;
- until the insertion point is found;
- if insertion is at the beginning, link the new node to the head and make the head equal to the new node;
- if insertion is at the end, link the previous node to the new node;
- copy the *next* pointer of the node previous to the insertion point to the *next* pointer of the new node;
- link the previous node to the new node;

Note: Allow the use of diagrams to illustrate any of the above points
