

1.4 DATA TYPES & STRUCTURES · 1.4.2

Lists & linked lists

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

Instructions. Answer all questions. Marks are shown in brackets []. Diagrams may be drawn in the spaces provided.

1 Total: 6 marks

A dynamic list `members` supports the usual list operations.

(a) State what the `append(item)` operation does. [1]

(b) Explain the difference between `pop()` and `remove(item)`. [2]

.....

(c) A list is initialised as `['Ada', 'Bea', 'Cody', 'Dia', 'Eli']`. Show the contents of the list after the following operations are carried out in order: [3]

`pop()`, `remove('Cody')`, `append('Finn')`, `pop(0)`, `append('Gus')`,
`insert(1, 'Hana')`

.....

2 Total: 7 marks

This question is about linked lists.

(a) Describe what is stored in each node of a linked list. [2]

.....

(b) State what the `start` pointer holds. [1]

(c) State what marks the end of a linked list. [1]

(d) Write a pseudocode algorithm to traverse the linked list and print every item's data. [3]

.....

3

Total: 6 marks

A linked list holds names in alphabetical order: Banana → Lemon → Melon → Strawberry.

- (a) Describe the steps needed to insert the name *Blueberry* into the linked list so the alphabetical order is maintained. [4]

- (b) Describe how the node *Melon* would be removed from the linked list. [2]

4

Total: 5 marks

- (a) Describe **two** differences between an array and a linked list. [4]

- (b) State one reason why inserting an item into the middle of a linked list can be faster than into the middle of an array. [1]

5

Total: 6 marks

A programmer writes a loop that goes through a list by index and removes every value over 100. The program crashes with an "index out of range" error.

```
for index = 0 to length(list) - 1
  if list[index] > 100 then
    list.remove(list[index])
  endif
next index
```

- (a) Explain why the program crashes. [3]

- (b) Describe one way to fix the algorithm so it removes the values correctly. [3]

END OF QUESTIONS