

1.4 DATA TYPES & STRUCTURES · 1.4.2

Lists & linked lists — Mark scheme

30 marks · spec 1.4.2(b)(c)

AO key: AO1 = knowledge · AO2 = application. Accept any valid alternative pseudocode. Do not award the same point twice.

Q	ANSWER	AO	MARKS
1(a)	Adds the item to the end of the list. (1)	AO1	1
1(b)	pop() removes (and returns) the item at a position / the last item (1); remove(item) deletes the first occurrence of a given value (1). (2)	AO1	2
1(c)	['Bea', 'Hana', 'Dia', 'Finn', 'Gus']. 1–2 items correct = 1; 3–4 correct = 2; all correct in order = 3. (3)	AO2	3

Q	ANSWER	AO	MARKS
2(a)	The data item (1); and a pointer to the next node (1). (2)	AO1	2
2(b)	The position/index of the first node in the list. (1)	AO1	1
2(c)	A null pointer (in the last node). (1)	AO1	1
2(d)	Set a pointer to start (1); loop while the pointer ≠ null, printing the data (1); move to the next node each time (1). e.g. currentPtr = start while currentPtr ≠ null print(list[currentPtr].data) currentPtr = list[currentPtr].pointer endwhile. (3)	AO2	3

Q	ANSWER	AO	MARKS
3(a)	Place Blueberry in a free node (1); traverse from start keeping previous/current until current's data > Blueberry (i.e. Lemon), so it belongs after Banana (1); set Banana's pointer to the new Blueberry node (1); set Blueberry's pointer to Lemon (1). (4)	AO2	4
3(b)	Make the previous node (Lemon) point past Melon (1); i.e. set Lemon's pointer to Strawberry (Melon's next), bypassing Melon (1). (2)	AO2	2

Q	ANSWER	AO	MARKS
4(a)	Two differences, 2 marks each (point + contrast), max 4: array fixed size (1) vs linked list dynamic (1); array stored contiguously / allows direct index access (1) vs linked list traversed via pointers (1); array insert shuffles elements (1) vs linked list just repoints (1). (4)	AO1	4
4(b)	Only a couple of pointers change, rather than shifting all following elements along. (1)	AO2	1

Q	ANSWER	AO	MARKS
5(a)	Removing an item shortens the list (1); but the loop's upper bound / indices were based on the original length (1); so the index eventually points beyond the (now shorter) list → index out of range (1). (3)	AO2	3
5(b)	Build a new list of the values to keep (those ≤ 100) (1); append each kept value as you loop the original (1); use the new list as the result (do not delete in place) (1). (Accept: iterate over a copy, or loop backwards.) (3)	AO2	3

Total for paper: 30 marks