Final test “A” – answer key and grading guide

Partial credit may be given. In the fill-in pseudocode every unambiguous, correct answer is acceptable (e.g. `<>` or `!=` instead of `≠`, `null` instead of `Nil`). Do not deduct points for minor syntax details.

Task 1 – True or false

14 points

1	2	3	4	5	6	7	8	9	10	11	12	13	14
T	F	F	T	F	T	F	T	T	T	F	T	T	F

Reasons for the debatable cases: 2 – sequential access, O(n) · 5 – the sentinel element takes up space · 7 – only the indexes change, O(1) · 12 – degenerate tree (e.g. sorted insertion) · 14 – `equals` is needed.

Task 2 – Multiple choice

10 points

1	2	3	4	5	6	7	8	9	10
b	c	c	b	b	b	a	b	a	b

7: 1000 + (2 · 10 + 3) · 4 = 1092 · 9: 5 + (1 + 2) · 4 − 3 = 14 · 10: (6 + 3) mod 8 = 1.

Task 3 – Number of steps

10 points

a)	b)	c)	d)	e)	f)	g)	h)	i)	j)
O(1)	O(n)	O(n)	O(1)	O(n)	O(1)	O(log n)	O(log n)	O(n log n)	$O(2^n)$

Task 4 – Stack

7 points

Output: `ATAD ERUTCURTS` (5 points; if the order of the words or the space is wrong: 3 points)
In general: it **reverses every word** of the text separately; the order of the words and the spaces are kept. (2 points)

Task 5 – Queue

7 points

Output: `2 4 6 3 1 5` (5 points; with at most one wrong number: 3 points)
In general: the **Josephus problem** for k = 2: *n* people in a circle, every second one is eliminated; the output is the order of elimination, the last number is the survivor. (2 points; the description “it removes/prints every second element” also gets full credit)

Task 6 – Algorithms on paper

8 points

a) Insertion sort (1 point per step, 4 in total)

Step	1 (insert 2)	2 (insert 9)	3 (insert 1)	4 (insert 5)
Array	[2, 6, 9, 1, 5]	[2, 6, 9, 1, 5]	[1, 2, 6, 9, 5]	[1, 2, 5, 6, 9]

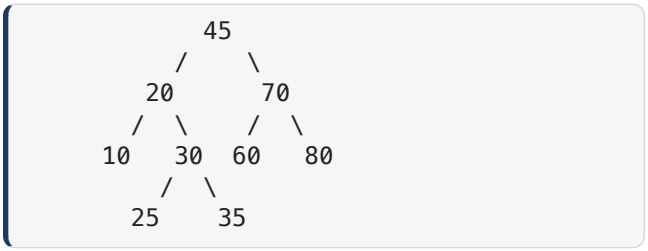
b) Binary search, x = 45 (1 point per step, 4 in total)

Step	left	right	k	A[k]	Decision
1	1	10	5	30	30 < 45 → left ← 6
2	6	10	8	52	52 > 45 → right ← 7
3	6	7	6	38	38 < 45 → left ← 7
4	7	7	7	45	found: index 7

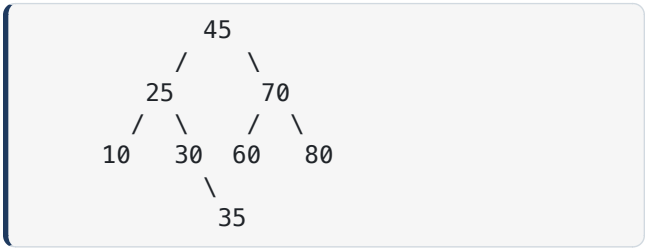
Task 7 – Binary search tree

12 points

a) The tree (3 points)



c) After deleting 20 (3 points)



b) Traversals (1 point each)

- Preorder: **45 20 10 30 25 35 70 60 80**
- Inorder: **10 20 25 30 35 45 60 70 80**
- Postorder: **10 25 35 30 20 60 80 70 45**

c) 20 has **two children** (case 3): it is replaced by the smallest element of the right subtree, the **inorder successor (25)**, then 25 (a leaf) is deleted. (naming the case 1 point, the correct tree 2 points; using the inorder predecessor (10) is also correct)

d) Height: **3** (45 – 20 – 30 – 25). Finding 35 takes **4** comparisons (45, 20, 30, 35). (height 1 point, comparisons 2 points)

Tasks 8–10 – Fill-in pseudocode

2 points per missing part

8. Search in a search tree (10 points)

	Answer
(1)	`Root`
(2)	`≠`
(3)	`≠`
(4)	`<` (`≤` is also acceptable)
(5)	`Elem[P].Right`

9. Sentinel search (10 points)

	Answer
(1)	`x`
(2)	`Head`
(3)	`≠`
(4)	`Elem[P].Next`
(5)	`=`

10. Insertion at the end of a list (12 points)

	Answer
(1)	`Nil`
(2)	`=`
(3)	`P`
(4)	`≠`
(5)	`Elem[Q].Next`
(6)	`Elem[Q].Next`

8/(4): if x = Elem[P].Key the loop does not run any more, so `≤` also gives the correct result. 9: thanks to the sentinel, the loop condition does not need the `P ≠ End` test. 10/(1): if the `Next` field of the new element is not Nil, the end of the list "dangles" – a common mistake.

Grade

Score	0–49	50–61	62–74	75–86	87–100
Grade	fail (1)	pass (2)	satisfactory (3)	good (4)	excellent (5)