11 Examinations - data structure finale exams

Algorithms And Data Structure (October University for Modern Sciences and Arts)

SECTION (I): TRACING QUESTIONS (Total: 7 marks)

(Q1) (3 mark)
- Draw the binary expression tree for the following prefix expression.
- Also, write the corresponding postfix, and infix expression.
- **+ + A * / * * B + K M C D W Y.**

(Q2) (4 mark)
For exercises (i - iii) use the following values, and draw the hash table for each of them:

      14 , 17 , 31 , 16 , 85 , 21 , 40

(i)   Store the values in a hash table with size 8 cells.
(ii)   Store the values in a hash table with 4 buckets, each bucket contains 2 cells.
(iii)   Store the values in a hash chain table with function (Key % 8).

(iv)   Fill in the following table, showing the number of comparisons needed to find each value:-

	(i)	(ii)	(iii)
21			
31			
40			

SECTION (II): ALGORITHMS (Total: 13 marks)

Algorithm 1: (5 marks)

Rewrite the **Selection-Sort** algorithm , but after each iteration check if the array becomes sorted or not, and return the number of iterations to sort the given array.
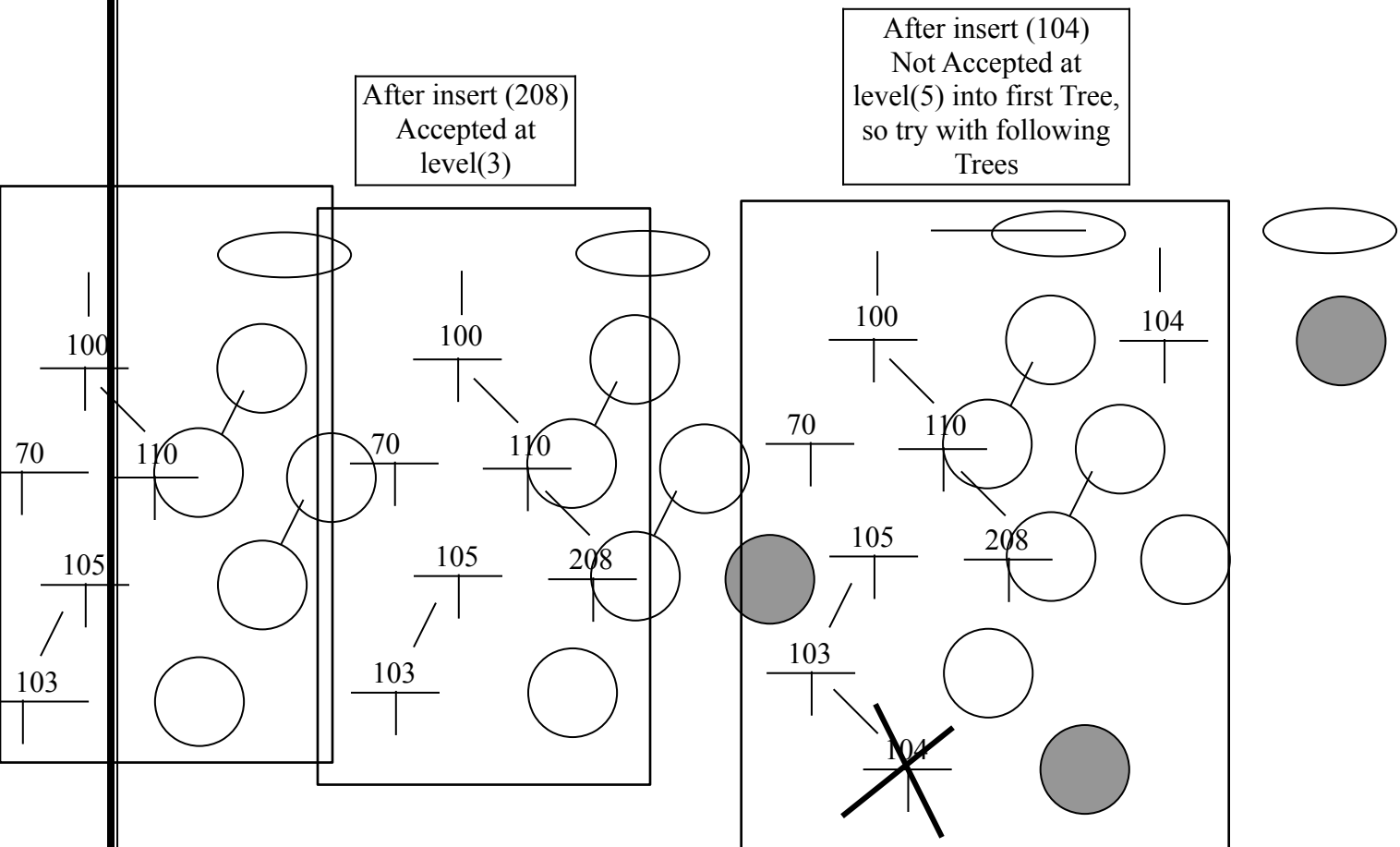
103	30	11	20	10	7	19

Algorithm 2: (8 marks)
You are asked to write a function called **Special_Insert (** *CTreeNode *pNN* **)**

- You will insert pNN in the first tree which accept your node according the following rule:
    - If the level of the inserted node will not exceed the Maximum height (**N**), so the node will be accepted.
    - You are responsible to declare the CTree and CListNode.
    - Assume that (N) already declared in the CTree class, and its value already assigned in the constructor.

    e.g.
    N = 4

After insert (208)
Accepted at
level(3)

After insert (104)
Not Accepted at
level(5) into first Tree,
so try with following
Trees

SECTION (III): PROBLEM SOLVING (Total: 20 marks)

Problem 1: (5 marks)

In the this question use the following definition for the nodes in the List:
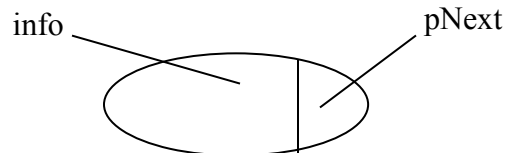
class CListNode
{
public:
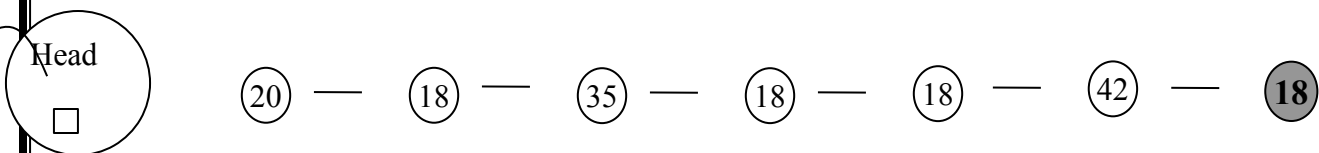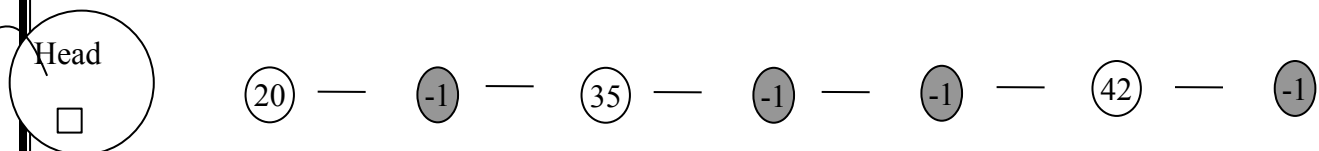   **int   info;**
      CListNode   *pNext;
};

info   pNext

Write a **recursive** function to replace all nodes which equal to the last node to be (-1)

e.g.

Head □

(20) — (18) — (35) — (18) — (18) — (42) — **(18)**

AFTER YOUR FUNCTION:

Head □

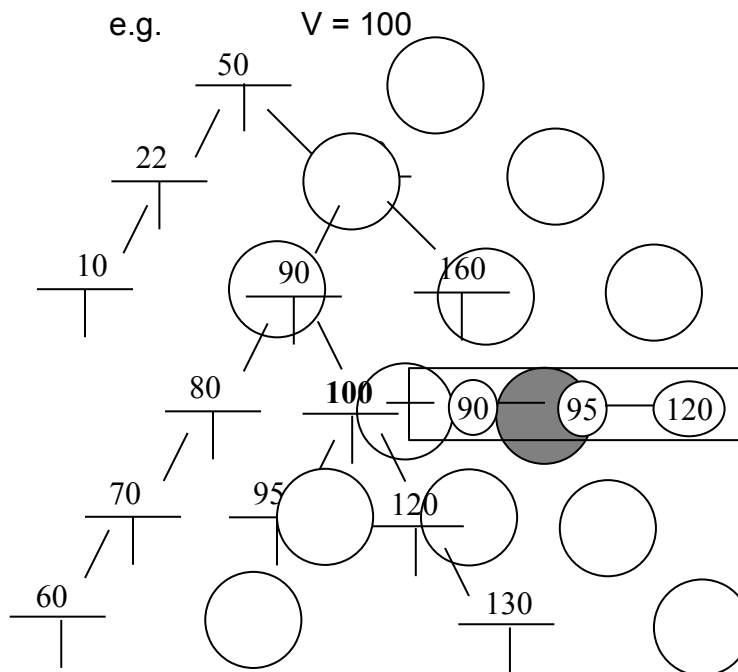(20) — (-1) — (35) — (-1) — (-1) — (42) — (-1)

Problem 2: (15 marks)
Write a main function to do the following:

(1) Read a Binary Tree from the user (T).
(2) Ask the user to select a target value (V), Find it and create an embedded Linked List that carries (the parent's value and the children's values).

    Note: declare the data structure of your CTreeNode

e.g.   V = 100

50

22

10

90

160

80   **100**   90   95   120

70   95   120

60   130

e.g.   V = 70

50

(3)   Repeat step 2 for N-time

(4)   Ask the user to select 2 nodes in the Tree, find and locate them.

(5)   Swap their embed   sts. You have to do this task in **O(1)**.

(6)   Display the embe   sts for the entire Tree.

22

10   90   160

80   100   90 — 95 — 120

60 — 80   **70**   95   120

60   130