CSI2131, Winter 2002
Lab 11

Question 1. B trees: insertions and deletions

V1) Variation using last key of a child (as in Chapter 9)
This is the standard way B trees were introduced in Chapter 9.

The format of a node is exemplified by the following node, which appeared in Figure 9.22 in page 406 of the textbook as the root of the B-tree:

	E		N		U		Z		-

V2) Variation using first key of a child.
This is a variation in which the first key of the child (rather than the last key) is stored in its parent.

The format of a node is exemplified by the following node, which shows how the root of the B-tree in Figure 9.22 would be modified to store the first key of its children:

A		K		P		V		-	

In this question you are going to operate on B trees of **order 3**.
Start with an empty tree. Insert the following keys into the B-tree:
AD, CA, ER, FA, FR, AY, BO.
Show your tree only after the underlined insertions.
Delete CA and show your tree.

You are going to do the above tasks in two different ways:

A) Employ variation **V1** and use **merging** when deleting.

B) Employ variation **V2** and use **redistribution** when deleting.

Question 2. B-Trees: Insertion/Deletion

Consider the following three **B-Trees** of **order 4**:
(in case you get confused with the graphics: each key is a single letter)

```
B-Tree # 1:
-----------
            O    W
           /      \
          /        \
     G  M  O      Q  W
    /   |  |      |  \
   /    |  |      |   \
 BEG  JKM  NO    PQ    VW


B-Tree # 2:
-----------
           G  M   Q    W
          /   |    \    \
         /    |     \    \
        /     |      \    \
      BEG    JKM    NPQ   VW

B-Tree # 3:
-----------
           G  O  Q    W
          /   |   \    \
         /    |    \    \
        /     |     \    \
      BEG   JMNO    PQ   VW
```

Choose the correct answer:

1. B-Tree # 1 is obtained by adding key O to B-Tree # 2
2. B-Tree # 3 is obtained from B-Tree # 2 by doing the following in this order: adding key O and deleting key K (using a re-distribution strategy, if necessary).

3. B-Tree # 3 is obtained by deleting key K from B-Tree # 1 and using a re-distribution strategy to fix the tree.

4. B-Tree # 1 is obtained by adding key K to B-Tree # 3

5. B-Tree #2 is obtained from B-Tree # 3 by doing the following in this order: Deleting key O (using a re-distribution strategy, if necessary) and adding key K.

Question 3. B-Trees: Deletions

Consider the following B-tree of **order 4**:

	D		G		L		-

	A		C		D		-

	E		G		-		-

	I		J		L		-

When deleting key `G`, there are several possible resulting trees, depending on whether the algorithm employs merging or redistribution, and on which sibling is involved in the operation.

Each of the following nodes, except one, is a root of one of the possible resulting trees. Select the one that **is not** a root of one of these trees:

1.

	E		L		-		-

2.

	C		E		L		-

3.

	D		I		L		-

4.

	D		L		-		-

5.

	D		E		L		-

Question 4. B-Trees: Worst-Case Search Depth

What is the **minimum** number of keys that can be stored on the leaf level of a B-Tree of order 20 with 3 levels?
(Note: the root of a tree counts as its first level)

1. 8,000
2. 800
3. 1,000
4. 200
5. 2,000