1. Suppose that we have an ordered file with 30,000 records stored on a disk with block size 1,024 bytes. File records are of fixed size with record length 100 bytes.

a) Find the number of block accesses required to search for a record using a binary search.

b) Suppose that the search key field of the file is 9 bytes long, a pointer is 6 bytes long, and a primary index is constructed for the file with one index entry per data block. Find the number of block accesses required to search for a record using the index.

c) How many levels are required to construct a multilevel index on the primary index in b) such that there is only one index block at the top level? Find the number of block accesses required to search for a record using the multilevel index.

2. Consider a disk with block size 512 bytes. Suppose that the search key field of a file is 9 bytes long and a pointer is 6 bytes long. We want to construct a B^+-tree index for the file and a node of the B^+-tree is made to be the same size as a disk block.

a) What is the largest integer value of n for the B^+-tree?

b) What are the largest and the least number of search key values that can be stored at the leaf level of a 4-level B^+-tree?

3. Consider a B^+-tree and a given function `find(V)`, which returns leaf node C and index i such that $C.P_i$ points to the record with search key value V if such a record exists. Write a pseudocode for a procedure `printRange(L,U)` to find and print all records with search key values in a specified range (L, U), assuming both L and U exist in the tree and the number of keys in a leaf node is known. Such queries are called ***range queries***.