# Implementation of DBMS
# Exercise Sheet 6
# Klingemann, WS 2024 / 2025

1) We want to represent physical addresses for a hard disk. For a block address we need to identify the following entities: the cylinder, the track within a cylinder, and the block within a track. To each of these entities we allocate one or more bytes to identify it. Our disk has the following properties:

- 8192 cylinders
- 8 tracks in a cylinder
- 32 blocks in a track

a) How many bytes do we need for a block address?
b) We want construct a record address by adding the position of the byte within a block to the block - address of exercise a). The blocks of the disk consist of 4096 bytes How many bytes would we need for the record address?

2) Suppose that blocks can hold either ten records or 100 key-pointer pairs. We have a data file that is a sequential file and a sparse index on this file. The index has multiple levels up to a level with just one block. Each primary block of the data file has one overflow block. The primary blocks are full, and the overflow blocks are half full. However, records are in no particular order within primary block and its overflow block. All index blocks are 60% full.
a) Calculate the total number of blocks needed for a 3,240,000-record file and the index.
b) Calculate the average number of disk I/O's needed to retrieve a record given its search key by using the index. You may assume that nothing is in memory initially, and that the search key is the primary key for the records.

3) In a B+-tree (will be discussed later in the lecture) of order n, the minimum number of keys in a node can be calculated with the following formulars:

- non-leaf node: $\lceil (n+1)/2 \rceil - 1$
- leaf node: $\lfloor (n+1)/2 \rfloor$

What is the minimum number of keys in B+-tree (i) interior nodes and (ii) leaves, when
a) n = 10
b) n = 11