Homework 2
Due: February 11
For the following problems, assume that a character is stored in 1 byte, an integer is stored in 4 bytes and a real number is stored in 8 bytes.

1. Each record in your file is to hold the following information:
a. a 10 digit phone number
b. an hourly salary
c. a employee ID number (7 digits)
d. a delete field
e. the number of years of service of the employee (as a whole number)

Suggest an appropriate record structure assuming that a block is 1 Kb .
2. Each record in your file will hold a product ID (12 bytes long) and up to 8 associated parts (each 24 bytes long). Each product has at least 2 parts and no more than 8 . The probability that you will have 2 parts for a product is $40 \%, 3$ parts is $20 \%, 4$ parts is $5 \%, 5$ parts is $15 \%, 6$ parts is $5 \%, 7$ parts is $10 \%$ and 8 parts is $10 \%$.

Given that a block is 128 bytes long, compare the average number of blocks per record if you use each of the following record organizations (make sure you show all work don't forget about padding):
a. The record is variable length. It contains the product ID and each part associated with it. The record also contains its length.
b. The record is fixed length with enough space for all 8 parts.
c. The record contains the product ID and a pointer to a linked list of parts which is in a separate file. Do not forget to add the space needed for the linked list into your calculations.
d. Each record contains the product ID and $p$ parts (where p is a value between 1 and 7). There is a pointer to a linked list of parts in a separate file for when there are more than $p$ associated with the product ID. The pointer is nil if the linked list is not needed. Give the average number of blocks per record for each value of $p$. What is the optimal value of $p$ ?
3. We have a record that contains fields $A, B, C, D$. The probability of searching on the fields is, respectively, $1 / 2,1 / 4,1 / 8$ and $1 / 8$. Queries are written using exactly one of the fields. Given 2048 buckets, what is the optimal distribution of bits in the bucket address if we wish to use partitioned hashing?
4. We have a record that contains fields $A, B, C, D$. The probability of searching on the fields is, respectively, $8 / 9,1 / 2,1 / 9$ and $1 / 17$. Queries are written using one or more of the fields. Given 2048 buckets, what is the optimal distribution of bits in the bucket address if we wish to use partitioned hashing?
5. Let us define "complete indexing" to mean that an index exists for every distinct (and distinctly ordered) field combination in the indexed file. For example, complete indexing for a file with two fields $A$ and $B$ would require two indexes: one on the combination $A B$ (in that order) and one on the combination $B A$ (in that order). How many indexes are needed to provide complete indexing for a file defined on (a) 3 fields, (b) 4 fields, (c) $N$ fields?

