Searching a Database

Suppose that \( db \) is a database having the following structure:

- A database is a list of boxed items called records.
- A record is a boxed list of \( J \) items which are considered to be atomic (will not be further subdivided even though it might be possible to so do).
- Each record (a boxed list of \( J \) items) contains a key which is defined to be the contents (open) of the first item in a record.
- It is assumed that each record consists of two or more items, a key and one or more items.
Define a dyad \texttt{locate} which will return a one record database consisting of the first record having a key which matches the given key. The left argument of \texttt{locate} should be the database and the right argument should be the search key. If the database contains no records with a matching key, \texttt{locate} should return an empty database.

**Searching for all Records Matching a Key**

Define a dyad \texttt{locate\_all} which will return the database (possibly empty) of all records matching a given key. The left argument of \texttt{locate\_all} should be the database and the right argument should be the search key. If the database contains no records with a matching key, \texttt{locate\_all} should return an empty database.

**Discussion**

During the discussion of Laboratory Problem 4. we developed the idea of using an adverb

\[
\text{key} =: 1 : > @ (m@{ }) @ >
\]

...to be used for searches where the \texttt{key} field is not always the first item in a record. For example for the database \texttt{db}

```
> db
+-+-+-+-+-+-+
|1|2|0|4|3|
+-+-+-+-+-+-+
|0|1|2|3|4|
+-+-+-+-+-+-+
|0|4|1|3|2|
+-+-+-+-+-+-+
|3|1|4|0|2|
+-+-+-+-+-+-+
|0|2|1|4|3|
+-+-+-+-+-+-+
```

0 key db
1 0 0 3 0
0 = 0 key db
0 1 1 0 1

which provides useful functionality, but when the key items are not J atoms, such as is the case for the \texttt{people} database:

```
> people
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|howland|jack |computer science professor|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|clinton|bill |president usa |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|clinton|hillary|president usa |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|perot |ross |wanted to be president |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|bush |george |used to be president |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

0 key people
The key adverb fails because applying 0 key produces a function, which when applied, produces a result where shorter keys are padded on the right with the appropriate fill character for the type (space in the above example).

**Extra Credit Problem**

Define a conjunction, key which has a key-field number as left argument and key-search value as right argument which produces a function which embeds the matching on a key by key basis so that:

```
 0 key 'perot' "0 people
0 0 0 1 0
 0 key 0 "0 db
0 1 1 0 1
```