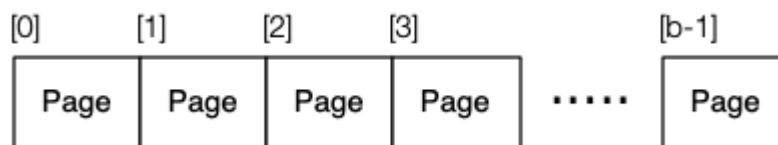


Background for Questions 1-3

The programming questions involve the manipulation of a data file type that I will call the "no-frills" file structure. Unlike the data files in Assignment 2, there is only a single data file; no info file, no overflow file.

Data File

A "no-frills" data file consists of a sequence of one or more pages; each page is `PAGESIZE` bytes long. All pages, except the last page, will be full (or almost full) of tuple data; the last page must contain one or more tuples. Even if the last page is not full of tuples, it will be `PAGESIZE` bytes long. Unused space in any page will be filled with `'\0'`. There should never be a page with zero tuples, except in the case of a file with a single page which is allowed to contain `PAGESIZE` bytes, all of which are `'\0'`.



A completely empty file (zero bytes) is *not* a "no-frills" file. A file whose size is not a multiple of `PAGESIZE` bytes is *not* a "no-frills" file.

Pages

Each page in the file has a single-byte "header" which simply contains a count of how many tuples are in the page.

| ntuples | 'a' | 'b' | 'c' | ',' | 'd' | 'e' | ',' | 'f' |
|--|-----|-----|------|-----|------|------|-----|-----|
| 'g' | 'h' | 'i' | '\0' | '1' | '2' | '3' | '4' | '.' |
| '5' | '6' | '7' | ',' | '8' | '\0' | 'e' | 'f' | 'g' |
| ',' | 'j' | ',' | 'k' | 'l' | 'm' | '\0' | etc | etc |
| Full of comma-separated, nul-terminated tuples Last tuple is terminated by two nuls (\0\0) or by the end of the page | | | | | | | | |

Tuples are added to a page buffer sequentially, and are simply appended after any existing tuples in the page buffer. When a page buffer does not have enough room to hold the next tuple, the current page is written at the end of the file. The page buffer is then cleared and is ready to receive more tuples.

Tuples

Tuples in "no-frills" files, are like the tuples in the data file in Assignment 2: a sequence of ascii characters, where attribute values are separated by a comma (','), and the tuple is terminated by a nul-character ('`\0`'). The last tuple in each page is terminated by two nul-characters. The first of these is its own nul terminator; the second of these indicates the end of tuple data.

Tuples contain at most `MAXTUPLELEN` bytes, including the commas and the terminating '`\0`'. Tuples can have 2-5 attributes, and each attribute value is a sequence of 2-10 alphabetic characters.

Debugging

Note that the data files, while consisting primarily of ascii characters, are still binary files. Examining them with `cat`, `less` or a text editor will not be helpful. The `od` command shows them in a usable format (i.e. text). You can use the `od` command as follows:

```
$ od -c data/Data1
0000000 024  o  a  t  a  v  t  d  v  ,  z  d  l  j  ,  g
0000020  n  u  m  g  o  ,  u  z  s  w  \0  c  u  w  c  k
0000040  x  y  m  u  ,  q  r  x  r  v  k  a  u  ,  n  q
0000060  f  u  g  t  o  ,  l  n  k  v  r  g  s  u  q  p
0000100  \0  j  n  u  d  m  n  y  ,  n  u  o  b  n  t  \0
... plus many more lines of output ...
```

If you want to compare two "no-frills" data files, you can use the `diff` command, but this only tells you whether the files are identical or not, e.g.

```
$ diff DataA DataB
Binary files Data1 and Data2 differ
$ diff DataA DataA
```

No output means that the files are identical.

If you want more human-understandable output of the differences, you could try something like:

```
$ od -c DataX > x
$ od -c DataY > y
$ diff x y
```

This may, or may not, give you some useful insight into the differences.