

COMP3190

Principles of Programming Language

Exercise 4 Answer

1. (Ignore)

2.

Boolean variables stored as single bits are very space efficient, but on most computers access to them is slower than if they were stored as bytes.

3.

Integer values stored in decimal waste storage in binary memory computers, simply as a result of the fact that it takes four binary bits to store a single decimal digit, but those four bits are capable of storing 16 different values. Therefore, the ability to store six out of every 16 possible values is wasted. Numeric values can be stored efficiently on binary memory computers only in number bases that are multiples of 2. If humans had developed hands with a number of fingers that was a power of 2, these kinds of problems would not occur.

4.

When implicit dereferencing of pointers occurs only in certain contexts, it makes the language slightly less orthogonal. The context of the reference to the pointer determines its meaning. This detracts from the readability of the language and makes it slightly more difficult to learn.

5.

The only justification for the `->` operator in C and C++ is writability. It is slightly easier to write `p -> q` than `(*p).q`.

6.

Let the subscript ranges of the three dimensions be named `min(1)`, `min(2)`, `min(3)`, `max(1)`, `max(2)`, and `max(3)`. Let the sizes of the subscript ranges be `size(1)`, `size(2)`, and `size(3)`. Assume the element size is 1.

Row Major Order:

```
location(a[i,j,k]) = (address of a[min(1),min(2),min(3)])  
+ ((i-min(1))*size(3) + (j-min(2))*size(2) + (k-min(3)))
```

Column Major Order:

```
location(a[i,j,k]) = (address of a[min(1),min(2),min(3)])  
+ ((k-min(3))*size(1) + (j-min(2))*size(2) + (i-min(1)))
```

7.

The advantage of this scheme is that accesses that are done in order of the rows can be made very fast; once the pointer to a row is gotten, all of the elements of the row can be fetched very quickly. If, however, the elements of a matrix must be accessed in column order, these accesses will be much slower; every access requires the fetch of a row pointer and an address computation from there. Note that this access technique was devised to allow multidimensional array rows to be segments in a virtual storage management technique. Using this method, multidimensional arrays could be stored and manipulated that are much larger than the physical memory of the computer.

8.

Implicit heap storage recovery eliminates the creation of dangling pointers through explicit deallocation operations, such as **delete**. The disadvantage of implicit heap storage recovery is the execution time cost of doing the recovery, often when it is not even necessary (there is no shortage of heap storage).