Kenneth Tran

CS3363

HW 6

2) Decimal values are design to be accurate to given number of decimal places. Because of this, decimal takes more memory than binary representation because decimal values must be represented by using at least 4 -bits per digit in the decimal number. However, it takes only 20 bits to store the same number in binary.

7) The operator → combines the features of dereferencing and field referencing. There are two ways of referring a field in a record using pointers. A pointer can be re-assigned any number of times while a reference can not be reassigned after initialization. A pointer can point to NULL while reference can never point to NULL. You can’t take the address of a reference like you can with pointers. There’s no “reference arithmetics”.

9) row major order:

**location (a[i,j,k]) = (address of a[min(a), min(b), min(c)]) + ((i-min(a))\*size(i)**

**+ (j-min(b)))\*size(ii) + (k-min(c))\*1)**

a[min(a), min(b), min(c)] a re the subscript of lower bound element.

i-min(a) will give number of rows between final and initial location

size(i) gives number of elements in each row

j-min(b) give number of col between final and initial location

k-min(c) are elements left in last axis

Column major order:

**location (a[i,j,k]) = (address of a[min(a), min(b), min(c)]) + ((k-min(c))\*size(iii)**

**+ (j-min(b)))\*size(ii) + (i-min(a))\*1)**

a[min(a), min(b), min(c)] a re the subscript of lower bound element.

k-min(c) will give number of axis between final and initial location

size(i) gives number of elements in each axis

j-min(b) will give number of cols between final and initial location

10) The main benefit of this scheme is that accesses that are done in order of the rows can be made extremely fast. After the pointer to a row is in hold, all the elements of the row can be fetched speedily.

The main disadvantage is that if the elements of a matrix have to be mandatory accessed in the column order, these accesses will be comparatively very slow.

20) Dynamic type checking is better than static type checking in way of a language allowing a memory cell to store values of different types at different times during execution. Such memory cells can be created with Ada variant records, C and C++ unions, and the discriminated unions of ML, Haskell, and F#. In these cases, type checking, if done, must be dynamic and requires the run-time system to maintain the type of the current value of such memory cells. So, even though all variables are statically bound to types in languages such as C++, not all type errors can be detected by static type checking.