Miguel Muniz
CS 326
Hw 5
4/12/2023

1.

A: array [0..9] of record
    s : short = 2 bytes
    c : char = 1 byte
    t : short = 2 bytes
    d : char = 1 byte
    r : real = 8 bytes
    i : integer = 4 bytes

2+1+2+1+8+4 = 18 bytes + padding of 4 bytes = 22 bytes
22 bytes * 10 = 220 bytes total need to store the array.

2.

Type T = array [1..10] of integer
     S = T
a : T
b : T
c : S
d : array [1..10] of integer

a) a b c and d have structural equivalence

b) a and b are structural name equivalent

c) a b and c are loose name equivalent

3) To fix the run-time error we need to fix the allocate function and the fact that the
    passed variable of q in the function would try to allocate a new pointer and overwrite itself.
    To fix this particular issue we need to pass the the Cell struct by reference and then allocate
    the memory to the pointer that is passed in.

code to fix the error:

```
void AllocateCell(Cell **q) {
  *q = (Cell *) malloc(sizeof(Cell));
}

int main() {
  Cell *c;
  AllocateCell(&c);
  c->a = 1;
  free(c);
  return 0;
}
```

4) Address of A[0][0] is 1000
the address of A[3][7] would be 1000 + [3]*10*8 + [7]*8 = 1000 + 240 + 56 = 1296

This is because the size of the int is 4 and size of the character c is 1 which makes the size of the strut to be 5. the virtual size of the strut is 8 because of the padding. so its the column * column size * 8 + the row * 8.