```
- Notes on 03-06-18
- begin in: pg. 73 in hymnal
```

Indexing Sequentially allocated structures

Arrays

Let's say we're working on the starship enterprise.

LOC(A(P,R,C)) LOC(A(Plane,Row,Column))

LOC(A(P,R,C)) = base + a1(p) + a2(R) + A3(C) ex at 1,1,2

```
base + 6(p)+3(r)+ col
100 + 6(1)+3(1)+2 = 111 === a(1,1,2)
```

- so:
    - a1 = rows*columns
    - a2 = number of rows
    - a3 = number of columns

LOC(A(P,R,C)) = base + [a1(p) + a2(R) + A3(C)] * C where '*C*' = size

A Using negative plane

new system A:(-1..0,0..1,1..3) new system = base + a1 (p+1) + a2(R) + a3(col-1) d1 = 2; d2 = 3; d3 = 3;

base + a1 (p+1) + a2(R) + a3(col-1)

100 + 6 (-1+1) + 3(1) + 1(2-1)

Let's make this *MUCH* larger
A(-10..10,20..30..red..green,0..100,green..purple,-10..5) A(I,J,K,L,M,N)

Bottom of page 74

LOC[A(i,j,k,l,m,n)] = BASE + a1(I+10) + a2(j-20) + a3(k) + a4(l)+ a5(m-green) + a6(n+10)

d1 = 21; d2 = 11; d3 = 3; d4 = 101; d5 = 3; d6 = 16;

a6 = 1; a5 = d6 = 16; a4 = d5 * d6 = 3 * 16; a3 = d4 * d5 * d6 = 101 * 3 * 16; a2 = d3 * d4 * d5 * d6 = ...; a1 = d2 * d3 * d4 * d5 * d6 = ...;

Loc[A(J,K)] := Loc[A(0,0)] + (J(J+1))/2 K