

ADT: data declaration  
+  
meaningful operations for the data type.

- implementation of these operations and storage of data may be hidden.

(encapsulation or data hiding)

ADT operations. - interface to public functions.

Ex. add-to-list (node) create-list ()  
list ADT delete-from-list (node) destroy-list ()  
find-in-list (node)

ADT implementation.

list ADT in 'C'.

(1) Array implementation.

list element  $\rightarrow$  unique key  $\rightarrow$  index into Array.  
(integer)

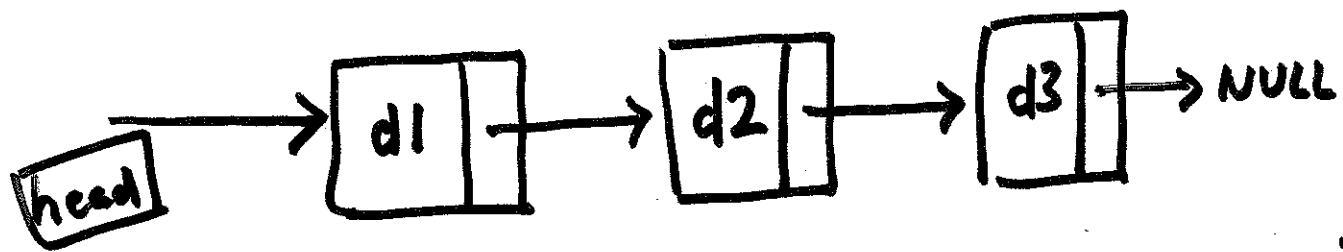
- searching could be easy (index)  $A[i]$

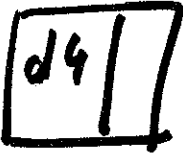
- insert and delete (complex)

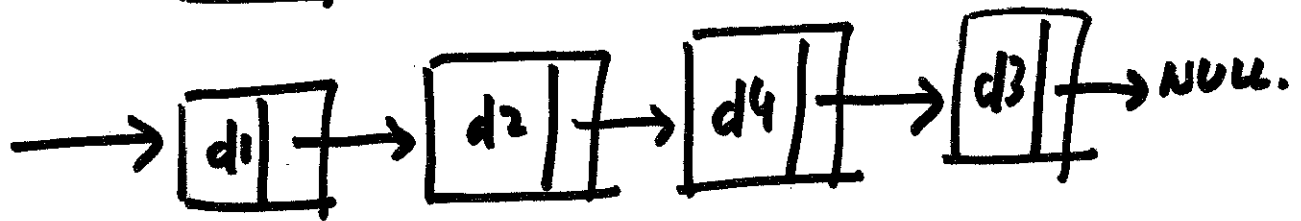
$A = \begin{bmatrix} 0 & [L] \\ \vdots & \vdots \\ 99 & [L] \end{bmatrix}$

Linked List Implementation.

Each element contains data and (1 or more) links.  
 $\swarrow$   
pointer to next element.



Insert  [insert in bet<sup>w</sup> d2 and d3]



- insert and delete easier.
- search sequentially. (inefficient)

Each element of the list is called a Node.  
 self referential structure.  
 [has a link (pointer) to another node]

To be able to handle, a list of integer. OR.  
 a list of float or  
 a list of char etc.

using a single ADT list. → Generic Code.

Features in C:

- void ptr
- pointer to functions.

void ptr. - can point to any data type.

void \* p;

int x = 5;

p = &x;

- must be cast before dereferencing

→. printf ( "Integer = %d \n", \*p );  
~~(int) p~~  
 \*(int \*) p

pointer to function.

name of the function.

"fun" is a pointer.

```
void fun(void)
{ ....
  return;
}
```

① int main (void)

② void fun (void)

→ void (\*f1) (void); ↓

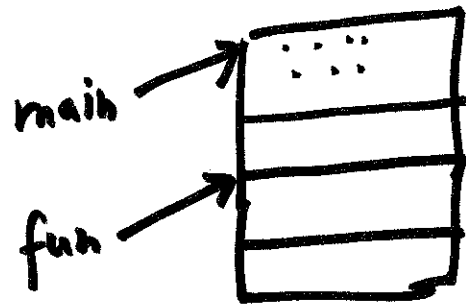
→ void \*f10(void);  
 ③ void \* f10(void);

③ int ~~fun~~ pun (int, int);

④ double sun (float);

f1 = fun; ✓

ADT: to find the largest value in the list?



## Application / User.

wants a list  $\rightarrow$  implementation.  
 $\rightarrow$  utilities.

implement a compare  
functions.

(e.g. compare.)

[node1  
node2]

- Larger (node1  $\rightarrow$  dataPtr,  
node2  $\rightarrow$  dataPtr,  
compare)

node1  $\rightarrow$  [wid # dataPtr.  
struct node # link.]

## ADT.

implement

- storage of list-  
[void #]
- maintenance of  
(linking) list
- utility f<sup>n</sup>.  
such as find...

[create Node  
- larger]