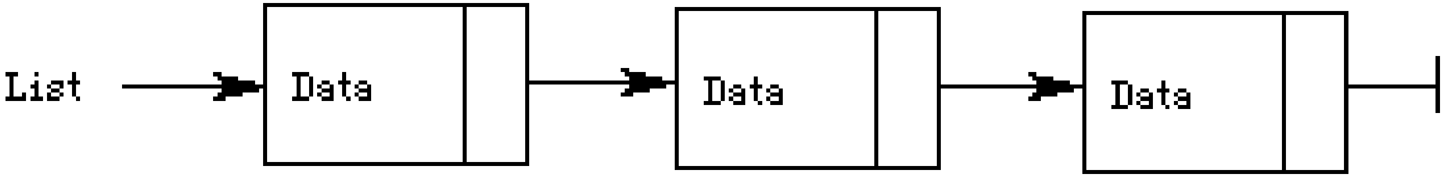
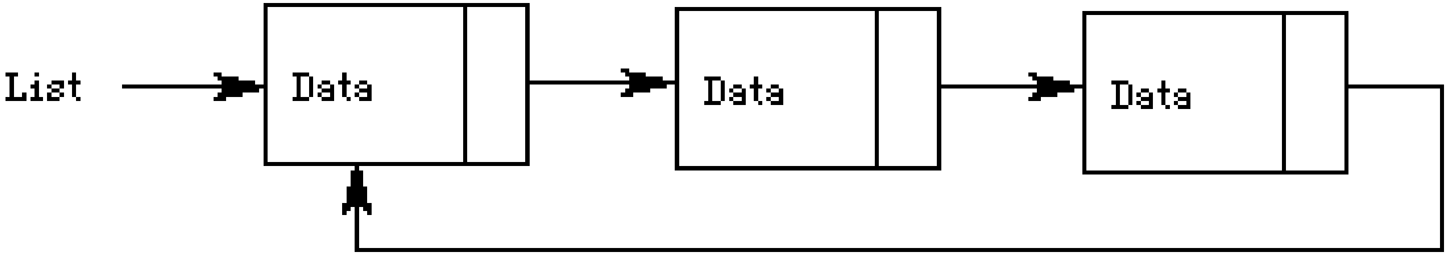
More Dynamic Memory

Linked Lists

Singly Linked List

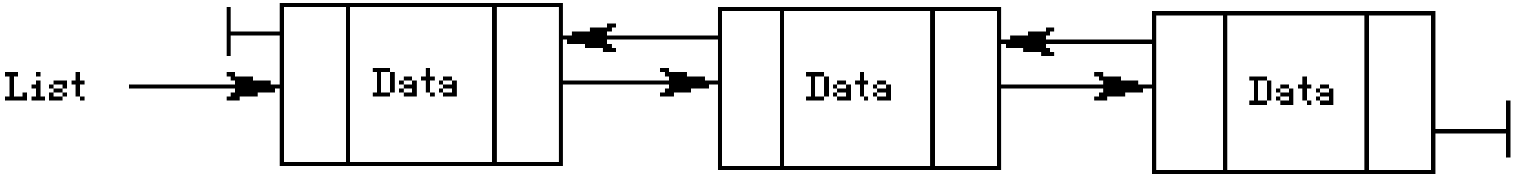


Singly Linked Circular List

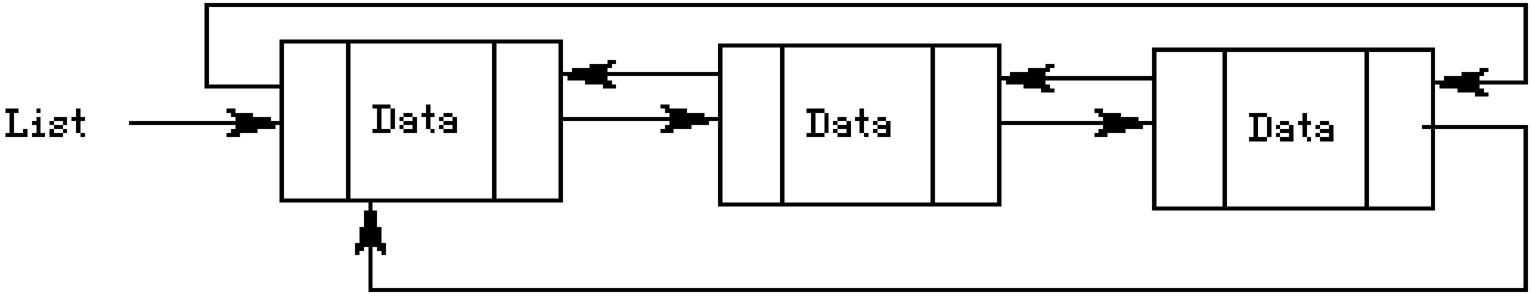


More Linked Lists

Doubly Linked List



Doubly Linked Circular List



What is a node?

```
struct NODE
       {
         int data;
         struct NODE* psNext;
       };

int main ()
{

  struct NODE sList;
  struct NODE *psList;

  return 0;
}
```

Which of these are legal?

```
sList.data = 5;
sList->psNext = NULL;
sList = NULL;
psList->data = 5;
psList = NULL;
```

Better C Definition for Node

```
typedef struct NODE *NODE_PTR;
typedef struct NODE
       {
         int data;
         NODE_PTR psNext;
       } NODE_ELEMENT;

int main ()
{

  NODE_ELEMENT sList;
  NODE_PTR psList;

  return 0;
}
```

Problems

- Create an empty list pointed to by **psList**.
- Allocate space for a new node and set the list pointer to point to the new node.
- Place the integer **10** into the data field of the single node.
- Create another new node and place the integer **20** into the data field of the new node.
- Link the two nodes together placing the node with 20 after the node 10.
- A linked list exists pointed to by the list pointer **psList**. Write a function **length** that accepts the list pointer to a singly linked list and returns the length of the list.