CS 241
Data Organization using C

Linked Lists

Instructor: **Joel Castellanos**

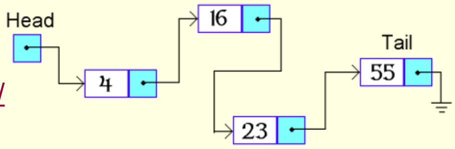
e-mail: joel@unm.edu

Web: http://cs.unm.edu/~joel/

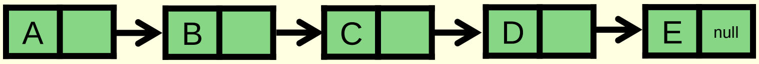
Office:

Farris Engineering Center

Room

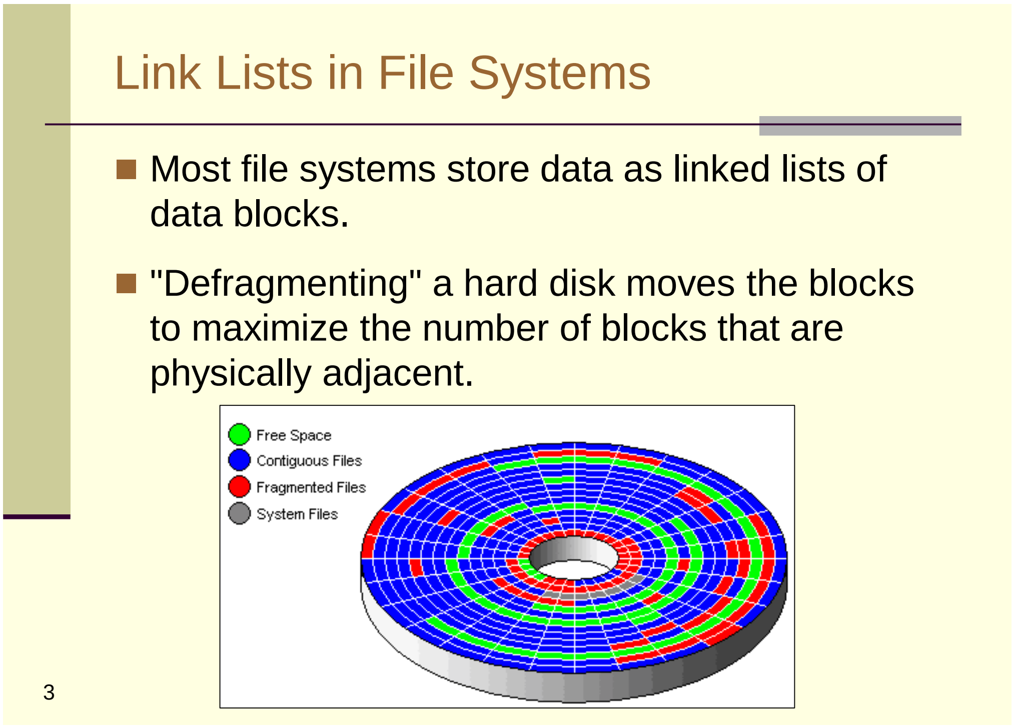


Linked List



A ***linked list*** is a *data structure* that consists of a sequence of data records such that in each record there is a field that contains a reference (a link) to the next record in the sequence.

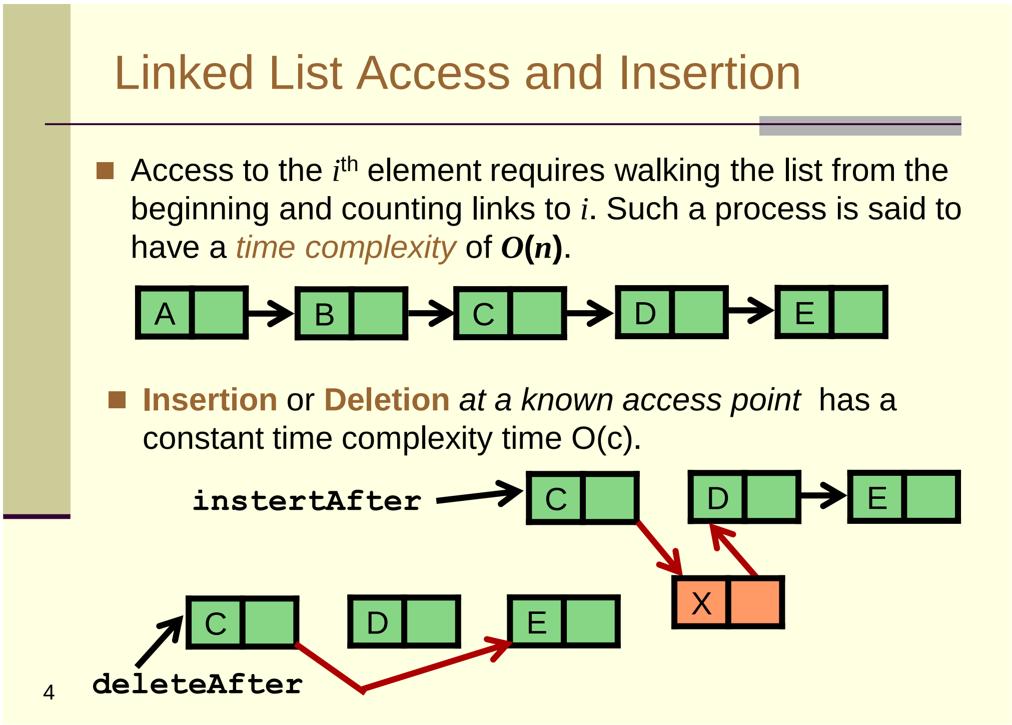
- The ***principal benefit*** of a linked list over a conventional array is that the order of the linked items may be different from the order that the data items are stored in memory or on disk.
- In the C programming language, the "link" is usually implemented as a pointer. The link could, however, be implemented in other ways (i.e. as an array index).



Link Lists in File Systems

- Most file systems store data as linked lists of data blocks.
- "Defragmenting" a hard disk moves the blocks to maximize the number of blocks that are physically adjacent.

🟢 Free Space
🔵 Contiguous Files
🔴 Fragmented Files
⚫ System Files



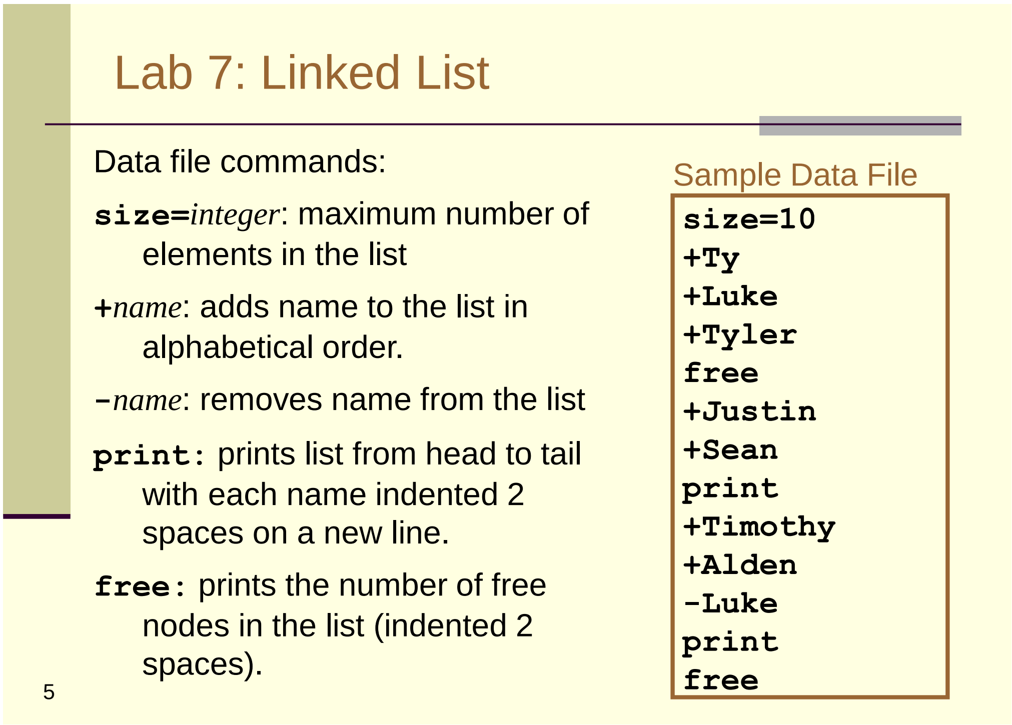
Linked List Access and Insertion

- Access to the i^{th} element requires walking the list from the beginning and counting links to i. Such a process is said to have a *time complexity* of $\boldsymbol{O(n)}$.

A
B
C
D
E

- **Insertion** or **Deletion** *at a known access point* has a constant time complexity time O(c).

instertAfter
C
D
E
X
C
D
E
deleteAfter



Lab 7: Linked List

Data file commands:

size=*integer*: maximum number of elements in the list

+*name*: adds name to the list in alphabetical order.

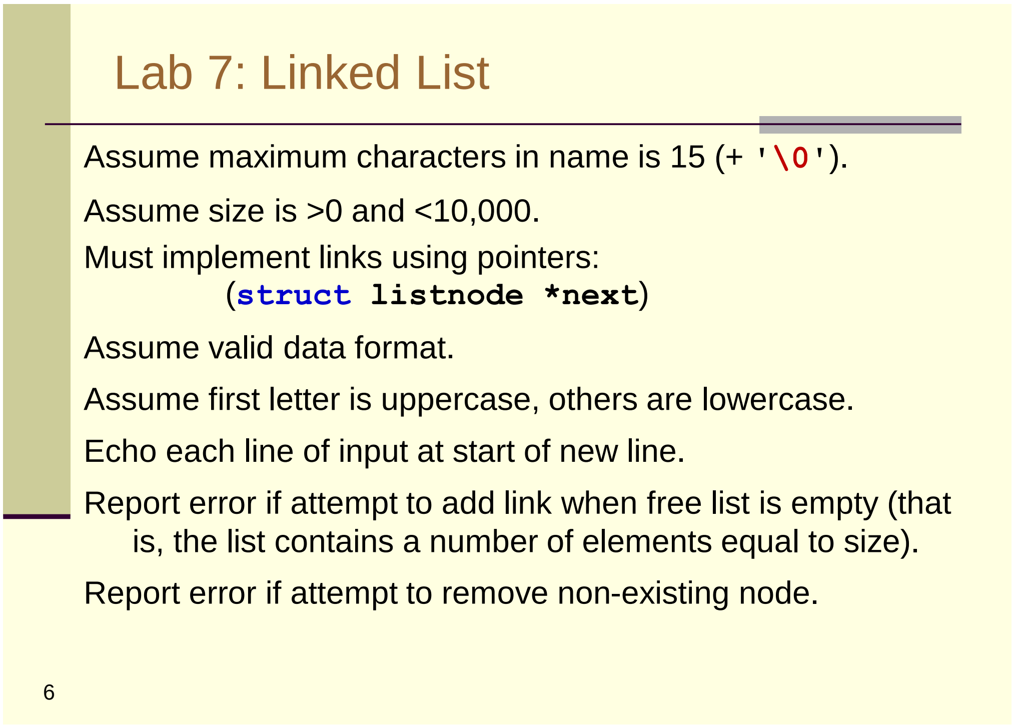
-*name*: removes name from the list

print: prints list from head to tail with each name indented 2 spaces on a new line.

free: prints the number of free nodes in the list (indented 2 spaces).

Sample Data File

```
size=10
+Ty
+Luke
+Tyler
free
+Justin
+Sean
print
+Timothy
+Alden
-Luke
print
free
```



Lab 7: Linked List

Assume maximum characters in name is 15 (+ `'\0'`).

Assume size is >0 and <10,000.

Must implement links using pointers:
(**`struct listnode *next`**)

Assume valid data format.

Assume first letter is uppercase, others are lowercase.

Echo each line of input at start of new line.

Report error if attempt to add link when free list is empty (that is, the list contains a number of elements equal to size).

Report error if attempt to remove non-existing node.

Linked List Structure (Suggestion, not Requirement)

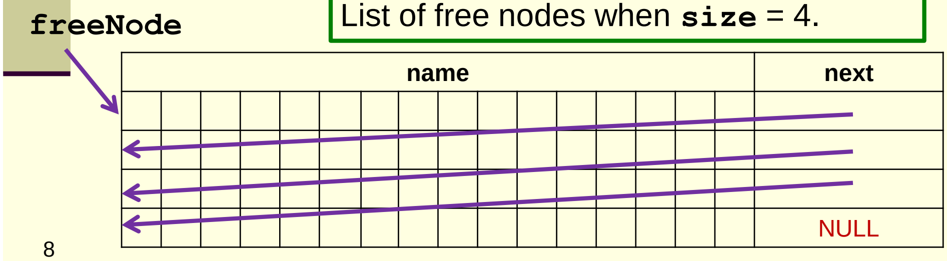
```
#define MAX_NAMES 10000
#define MAX_NAME_LEN 16

struct listnode
{ //Pointer to next node in chain
  struct listnode *next;

  char name[MAX_NAME_LEN]; //node data
};

struct listnode list[MAX_NAMES];
struct listnode *start = NULL;
struct listnode *freeNode;
```

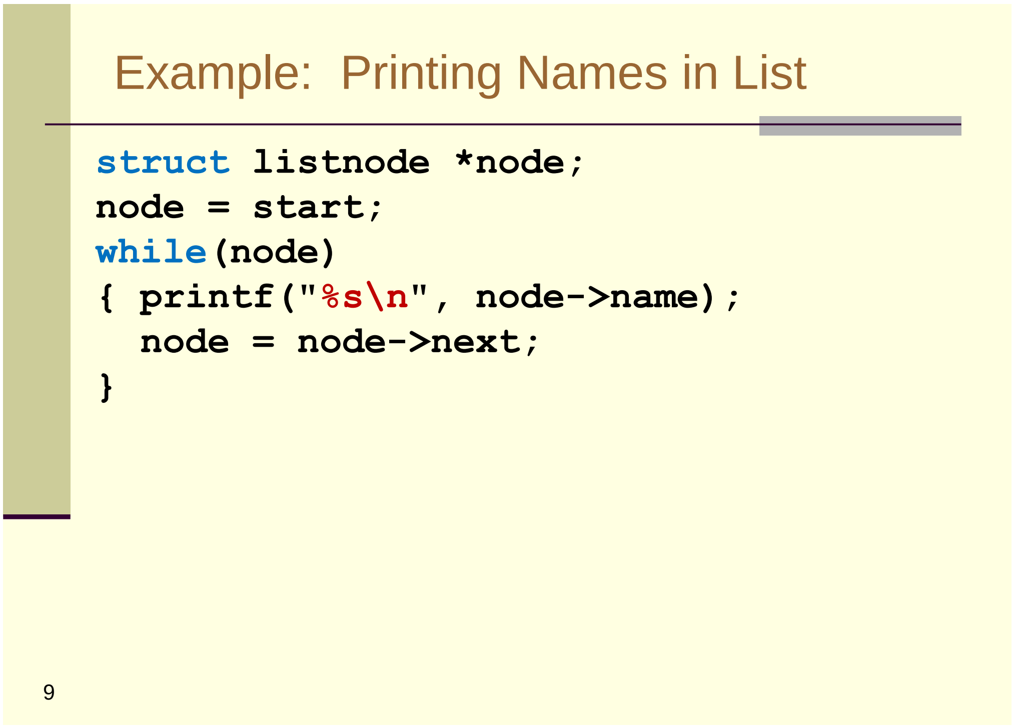
Initialize List of Free Nodes

```
freeNode = list;
int i;
for (i=0; i<size-1; i++)
{
  list[i].next = &list[i+1];
}
list[size-1].next = NULL;
```

size is input from the first line of the file.

List of free nodes when **size** = 4.



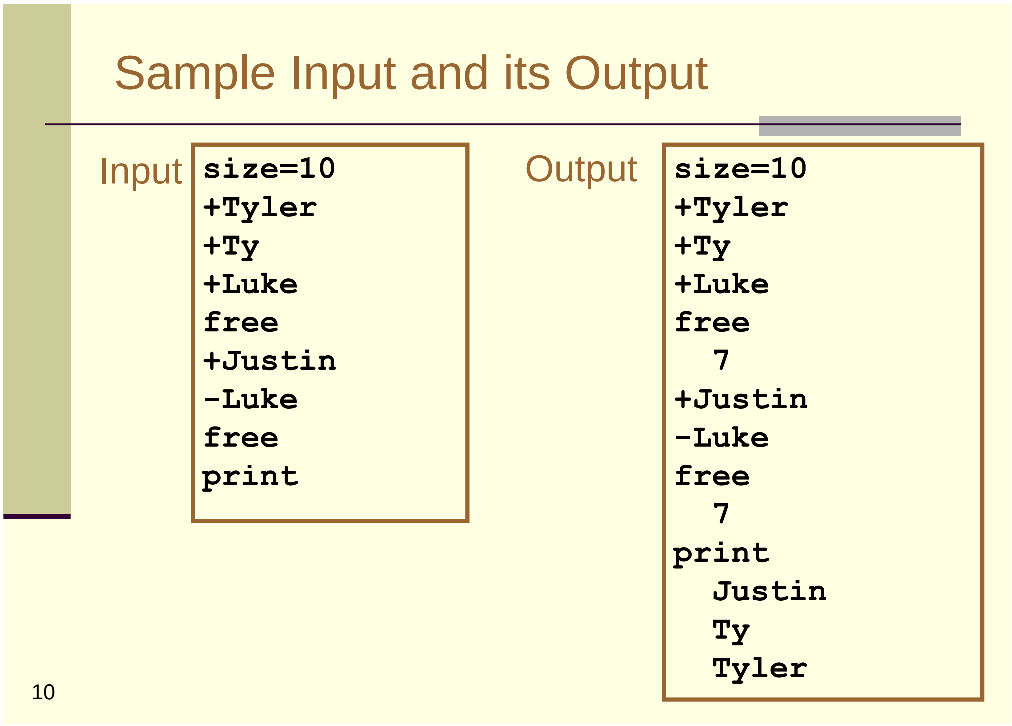


Example: Printing Names in List

```
struct listnode *node;
node = start;
while(node)
{ printf("%s\n", node->name);
  node = node->next;
}
```

9

9



Sample Input and its Output

Input

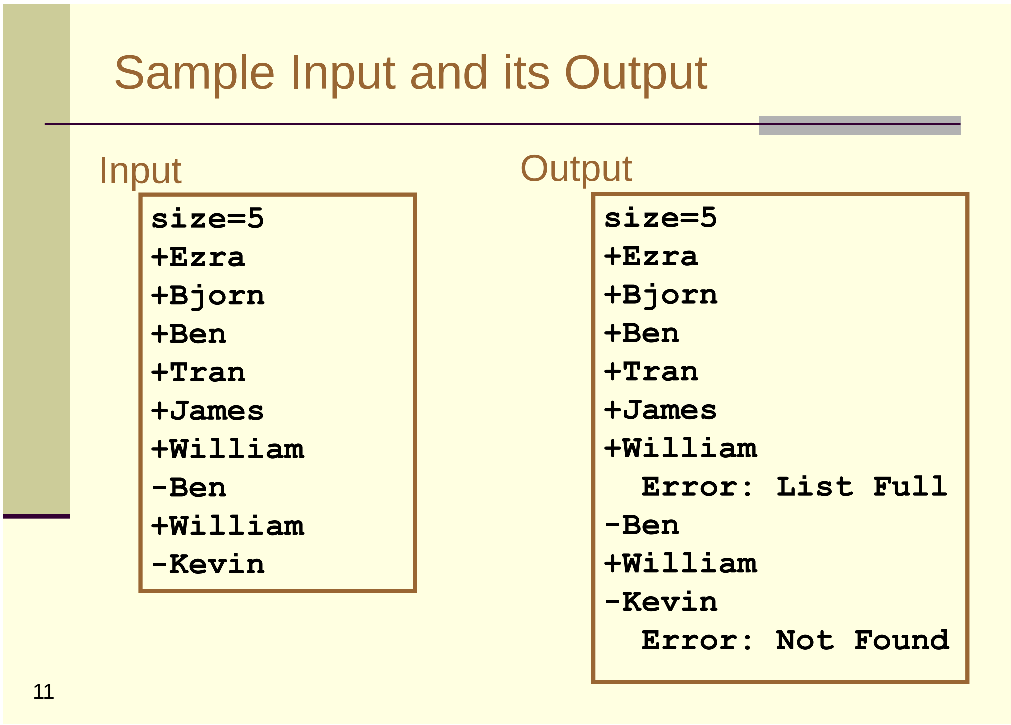
```
size=10
+Tyler
+Ty
+Luke
free
+Justin
-Luke
free
print
```

Output

```
size=10
+Tyler
+Ty
+Luke
free
  7
+Justin
-Luke
free
  7
print
  Justin
  Ty
  Tyler
```

10

10



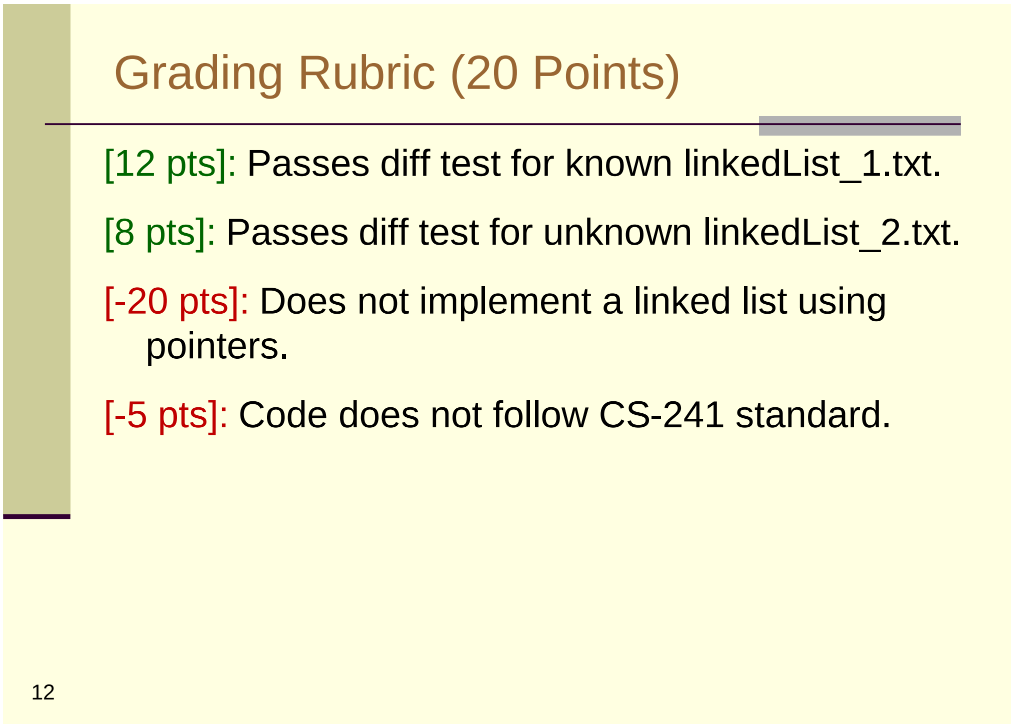
Sample Input and its Output

Input

```
size=5
+Ezra
+Bjorn
+Ben
+Tran
+James
+William
-Ben
+William
-Kevin
```

Output

```
size=5
+Ezra
+Bjorn
+Ben
+Tran
+James
+William
  Error: List Full
-Ben
+William
-Kevin
  Error: Not Found
```



Grading Rubric (20 Points)

[12 pts]: Passes diff test for known linkedList_1.txt.

[8 pts]: Passes diff test for unknown linkedList_2.txt.

[-20 pts]: Does not implement a linked list using pointers.

[-5 pts]: Code does not follow CS-241 standard.