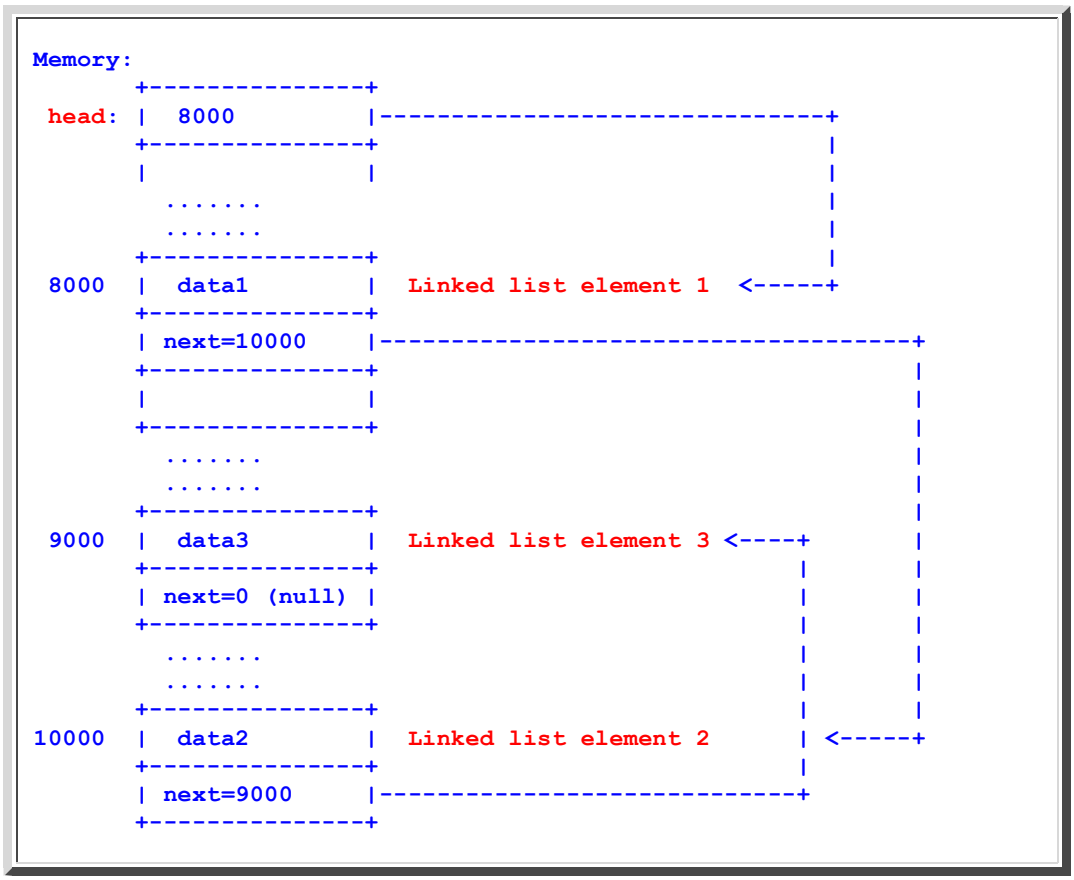
Traversing linked lists

- ***Recall*: linked lists**
  - **Recall** that a **linked list** is **chanied togeher** using **references**
  - **Example:** a list with **3 nodes**



- **Structure of our example list**
  - We use the following **Node** class definition in the **examples**:

```
public class Node        // Linked list element
{
   int value;    // 4 bytes
   Node next;    // 4 bytes
}
```

You will need to **adjust** the **offsets** for **different node structures**

The **underlying technique** on how to **traverse** the list will remain **unchange**

  - Suppose a **linked list** has been **constructed**; we just need to **traverse** the linked list.

The **list list** start at the **location** given by this variable:

```
Node head;
```

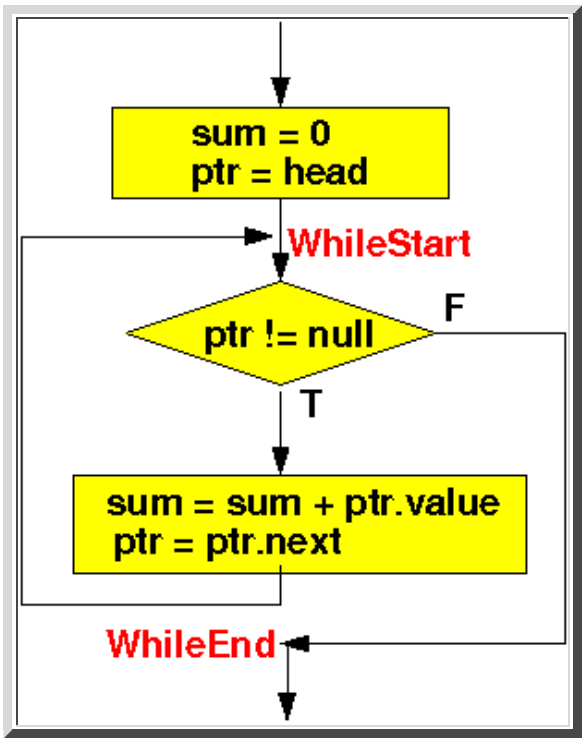
- **List traversal example 1: sum the values stored in a list**
  - The **Java** code to find the **sum** stored in all elements of the list:

```
int sum;
Node ptr;

sum = 0;
ptr = head;

while ( ptr != null )
{
   sum = sum + ptr.value;
   ptr = ptr.next;
}
```

The **flow chart** of the **above program** is:



In **M68000 assembler code**:

```
       move.l #0, sum
       move.l head, ptr

WhileStart:
```

```
        move.l ptr, d0
        cmp.l  #0, d0
        beq    WhileEnd

        movea.l ptr, a0
        move.l  0(a0),d0        * d0 = ptr.value
        add.l   d0,sum

        move.l  4(a0),ptr       * ptr = ptr.next

        bra    WhileStart

WhileEnd:
```

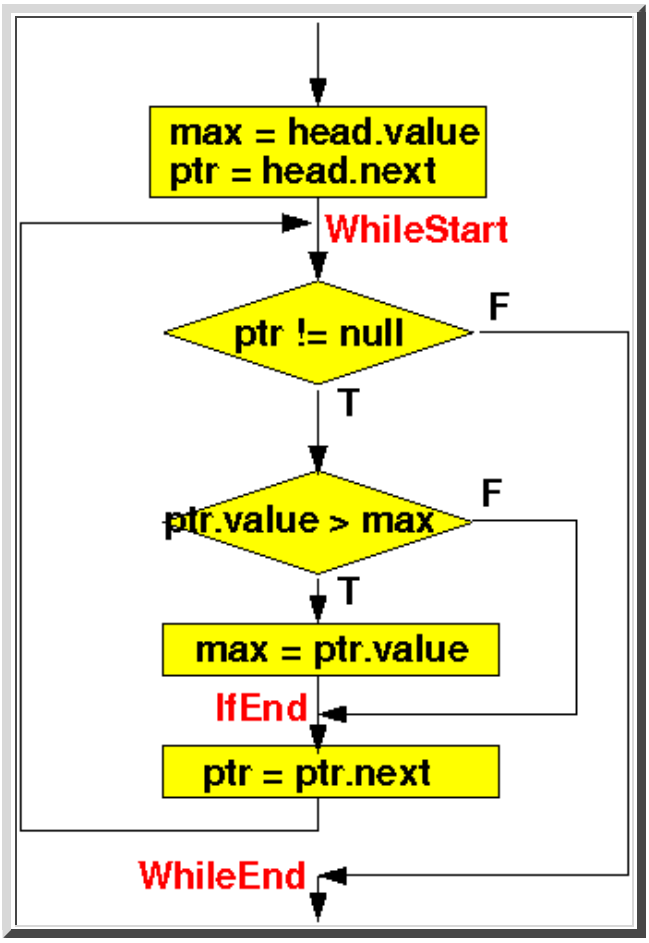
- Program: click here
  Egtapi debug file: click here

- **List traversal example 2: find max of the values stored in a list**
  - The **Java** code to find the **maximum** of the values stored in a list:

```
   max = head.value;
   ptr = head.next;

   while ( ptr != null )
   {
      if ( ptr.value > max )
         max = ptr.value;
      ptr = ptr.next;
   }
```

The **flow chart** of the **above program** is:



In **M68000 assembler code**:

```
      movea.l head, a0
      move.l  0(a0), max       * max = head.value
      move.l  4(a0), ptr       * ptr = head.next

WhileStart:
      move.l ptr, d0
      cmp.l  #0, d0
      beq    WhileEnd

      movea.l ptr, a0
      move.l  0(a0),d0         * d0 = ptr.value

      cmp.l   max, d0          * ptr.value - max
      ble     IfEnd            * If ( ptr.value - max <= 0 ) then branch

      move.l  d0, max          * max = ptr.value

IfEnd:
      move.l  4(a0),ptr        * ptr = ptr.next

      bra    WhileStart

WhileEnd:
```

- Program: click here
  Egtapi debug file: click here
