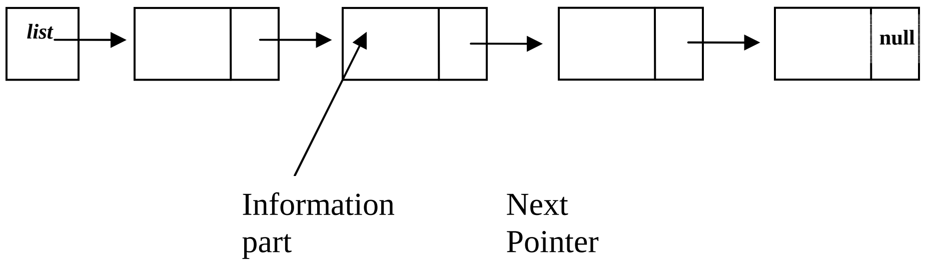
Computer Programming Lab **7/9/2007**

Linked Lists

Linked list, is a linear collection of data elements, called nodes, where the linear order is given by means of pointers. Each node can be divided into two parts: the first part contains the information of the element, while the second part contains the address of the next node of the list.

The figure shows a schematic of the linked lists. The left part contains the record field (which may be more than one fields also), while the right part represents the next pointer field. It points to the next node in the list. The pointer of the last node contains a special entry, called NULL (the null pointer). Ideally in C, we represent the data structure of linked list by self-referential structures (read from a standard book in C).



The advantages of such a data structure could be dynamic memory allocation, wherein we allocate memory only when we have a new element to be inserted into the list. Also we solve the problem of overflow when we have a fixed allocation of memory.

Write a C code to create a linked list to represent a polynomial of the form:

$f(x)=a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2}+ ... +a_0$

Each node represents each term. The information stored in each node should be the value of the coefficients and the value of the exponent. Assume that the coefficients are integers, while the exponents are whole numbers. While the next pointer looks into the node whose degree is just lesser than the current node.

Write insert and delete functions to insert any term into the function and also delete any term from the field. Use a separate print function to print the linked list after each operation. While printing each term is represented as: (a_i,i).