Array, List, and linked list
M.A. El-dosuky

[1] Explain different ways to store a collection of related data? **(arrays and linked lists)**
[2] What is the main disadvantage of an array?
[3] What is the main advantage of a linked list over an array?
[4] Discuss what is meant by linked list,

- **An array is a data structure where all elements are stored in contiguous memory;**
- **the defining parameters of an array are**
  - **Memory address of its First element (called the base address),**
  - **Size of each element (e.g. 4 bytes for an integer, 1 byte for a char, etc.),**
  - **Number of elements.**

Array advantages

- **Access time for any array element is constant**
  Location = base addresses + offset (index of required element)
- **Arrays allow random access to their elements**
- **Arrays require only as much space as is needed to store the data itself; no extra space for pointers or other markers is needed.**

➔ Array disadvantages

- **its size is fixed, either at compilation or when space is allocated dynamically**
- **Sufficient space must be allocated for the largest number of elements that will ever be used in the program, which can result in a lot of wasted space**
- **Certain algorithms are less efficient if used on data stored in arrays**

- **A list is a data structure in which the first element is stored on its own, but with a pointer to the location of the second object, which may be stored in a different area of memory.**
- **The second object in turn has a pointer to the third, and so on, until the last object has a null pointer associated with it to Indicate that it is the end of the list**

Advantages: Size need not be determined in advance, extra space can be allocated as needed.
Disadvantages: are that extra space is needed to store the pointer associated with each element, Access to internal list elements may take more time than with an array, since finding an internal list element requires starting at the first element and following the chain of pointers until the required element is located.

[5] Identify true or False and rewrite the correct version of False one.
A- An array is a Data Structure all elements are located in contiguous memory. **(true)**
B- The access time for any array elements is constant. **(true)**
C- In single dimension array, subscript can identify the number of individual elements in it. **(true)**
D- #define MONTHS 12
int array[MONTHS]; is equivalent to this statement: int array[12]. **(true)**
E- const int MONTHS = 12;
int array[MONTHS]; is equivalent to this statement: int array[12]. **(true)**
F- Dynamically allocated data Structure may be linked together to form a list **(true)**

[6] Define List ADT and its operations.

- **We will deal with a general list of the form A1, A2, A3,..., An. The size of list is N.**
- **We will call the special list of size 0 an empty list.**
- **For any list except the empty list, we say that Ai+1 follows (or succeeds) Ai, (i < N) and that Ai-1 precedes A, (i > 1).**
- **The first element of the list is A1, and the last element is An.**
- **We will not define the predecessor of Ai or the successor of An.**
- **Some popular operations are**
  1-PrintList
  2- MakeEmpty,
  3- Find, which returns the position of the first occurrence of a key;
  4- Insert and Delete, which insert and delete some key from some position in the list;
  5- FindKth, which returns the element in some position (specified as an argument).

Characteristics:

A List L stores items of some type, called ListElementType

Operations:

- **void L.insert(ListElementType elem)**
- **bool L.first(ListElementType &elem)**
- **bool L.next(ListElementType &elem)**

[7] What happens if I use

a subscript on an array that is larger than the number of elements in the array?

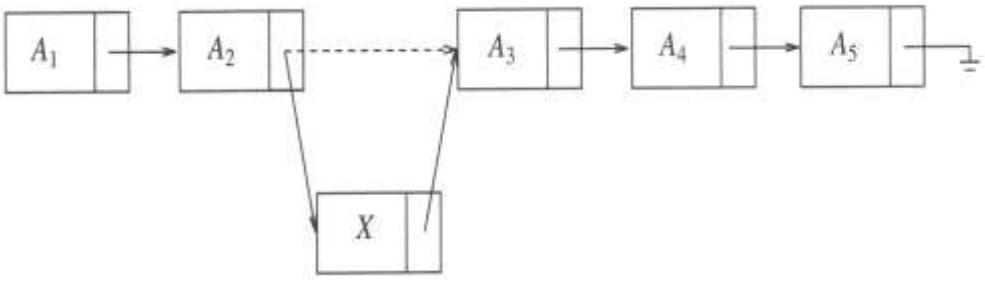
The program probably compile and run. The result of such mistake is unpredictable.

an array without initializing it?

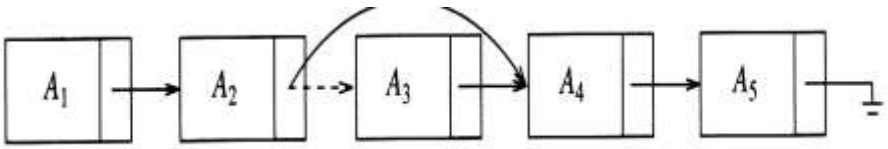
This mistake does not produce a compile error. Result of such mistake is unpredictable.

[8] Explain how to add and remove element from Linked List.

Insert requires obtaining a new cell from the system by using a malloc call and then executing two pointer maneuvers.



Delete can be executed in one pointer change. Result of deleting the third element in the list is :



[9] What are the basic linked list operations.

- **appending a node,**
- **traversing the list,**
- **inserting a node,**
- **deleting a node, and**
- **destroying the list.**

[10] Identify the error in the following programs and write the correct version of each one.

A- Program demonstrates use of an array to accept expenses data for 12 month and print them.

```
#include stdio.cpp
float expenses[13];
int count;
main
{
        for ( I = 1;     12; count++)
        {
        printf("Enter expenses for month %d: ", count++);
        scanf("%d", &count);
        }
        for (count = 0; count < 13; count)
                printf("Month %f = $%.2fin", count, expenses);
        return 0;
}
```

B- The program uses a single-dimensional array to store 10 grades and calculate the average.

```
include <stdio.h>
define MAX GRADE = 100
#define STUDENTS = 10
int grades[STUDENTS];
float idx;
int total = 0;
main()
{ for( idx = 0; idx < TOTAL; idx++)
        printf( "Enter Person %d's grade: ", idx -1);
        scanf( "%F", &grades[STUDENTS] );
        while ( grades[idx] MAX_GRADE )
        { print( "\nThe highest .grade possible is %d", MAX GRADE );
        printf( "\nEnter correct grade: " );
        scanf( "%F", grades[idx] ); }
        total = grades[idx];
        }
        printf( "The average score is %d\n", STUDENTS) );
        return (0); }
```

[11] A- How is a linked list declared in C++?

```
struct ListNode// Declare a structure for the list

    {
        float value;
        struct ListNode *next;
    };
ListNode *head;      // List head pointer
```

B- Describe the two parts of node.

- **The first is a float named value, used to hold the node's data**
- **The second is a pointer named next, and hold the address of any object that is a ListNode. This allows it to point to the next ListNode in the list.**

C-What is a list head? **The list head points to the first node in a linked list**

D-What signifies the end of a linked list? **NULL**

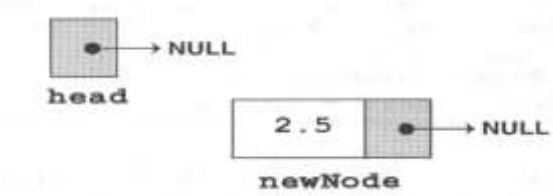
E-Self-referential data structure? **data structure that points to an object of same type as itself**

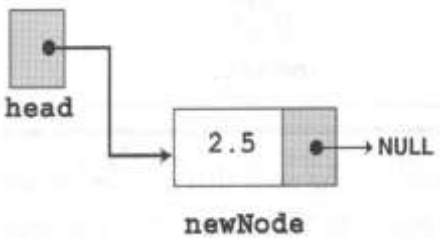
[12] Write the Pusedocode of algorithm that append a node to the linked list.

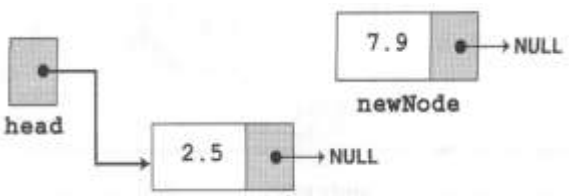
Create a new node.
Store data in the new node.
If there are no nodes In the list
Make the new node the first node.
Else
Traverse the list to find the last node.
Add the new node to the end of the list.
End If.

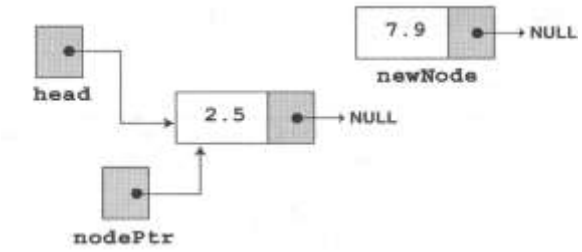
[13] Trace following program to append 2.5, 9.7, and 12.5 to draw the form of linked list.

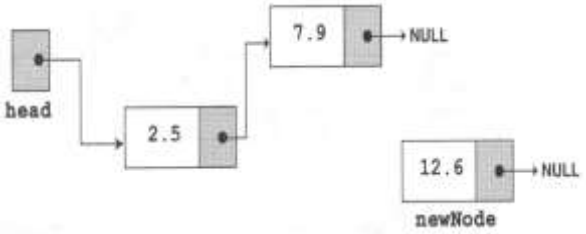
```
void main(void){
        FloatList list;
        list.appendNode(2.5);
        list.appendNode(7.9);
        list.appendNode(12.6);
}
```

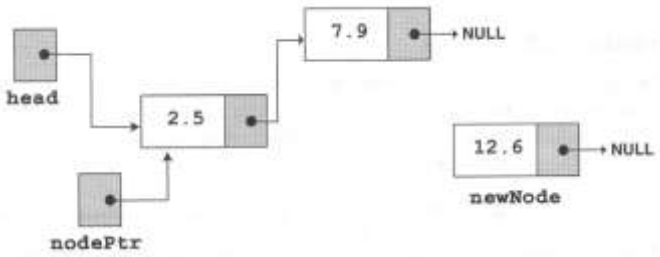


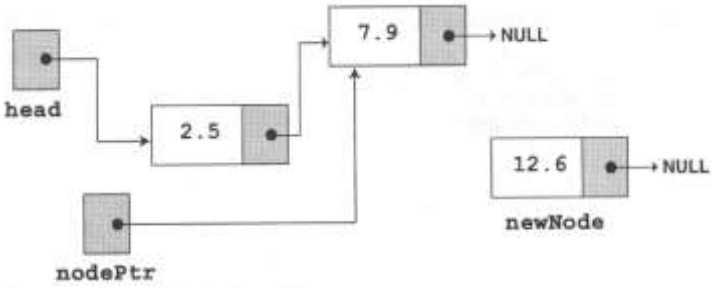


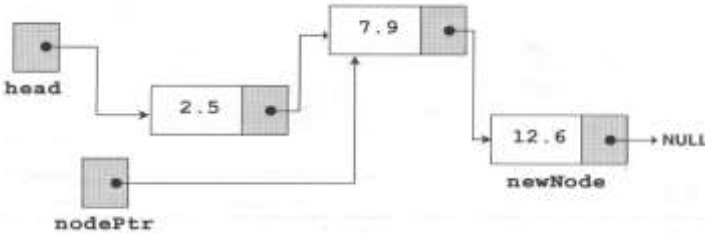






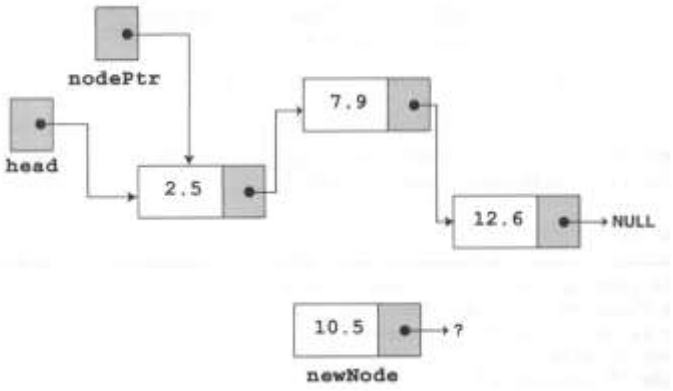


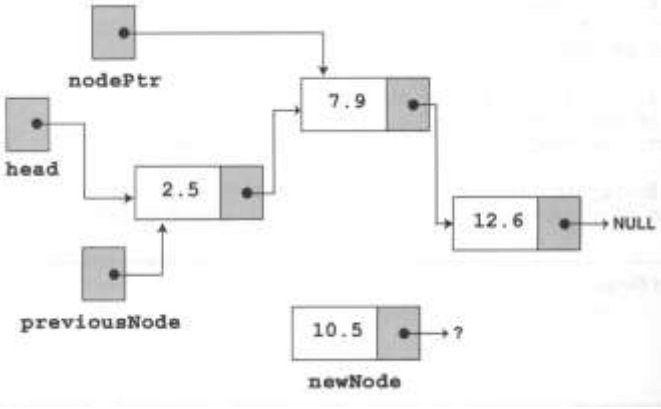


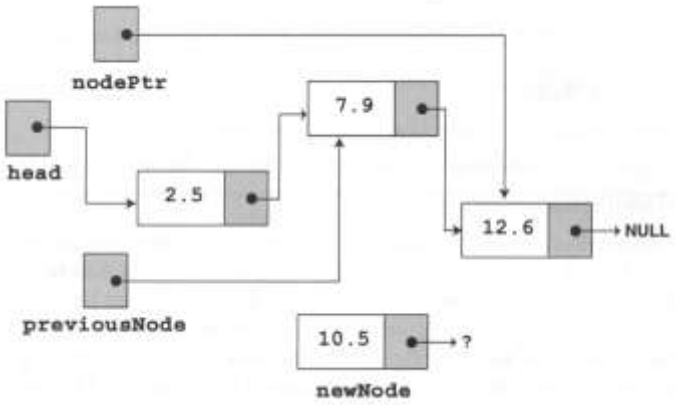


[14] What is output of the following program and draw the corresponding linked list structure.

```
void main(void)
{
        FloatList list; // Build the list
        list.appendMode(2.5); list.appendNode(7.9); list.appendNode(12.6) ;
        list.insertNode(10.5); // Insert a node in the middle of the list.
        list.displayList();// Dispay the list
}
```





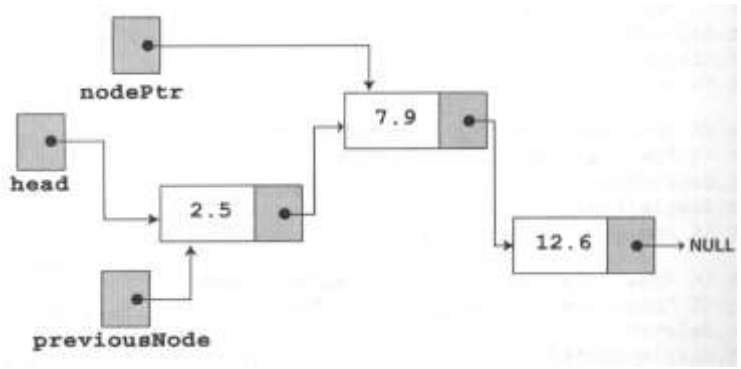


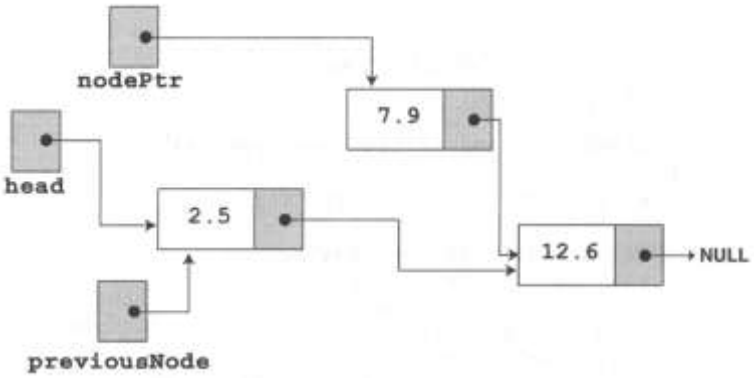
[15] write the algorithms of **Traversing and Deleting a Node**
And show **List.deleteNode(7.9);**

Traversing
displayList member function traverses the list, displaying value member of each node
Assign List head to node pointer.
While node pointer is not NULL
    Display the value member of the node pointed to by node pointer.
    Assign node pointer to its own next member
End While.

Deleting a Node
Deleting a node from a linked list requires two steps:

- **Remove the node from list without breaking the links created by the next pointers**
- **Deleting the node from memory (delete nodePtr)**





[16] Complete the following-:

a) The **head** points to the first node in a linked list,
b) A data structure that points to an object of the same type as itself is known as a **self-referential** data structure.
c) After creating a linked list's head pointer, you should make sure it points to **NULL** before using it in any operations.
d) **Append** a node means adding it to the end of a list.
e) **Insert** a node means adding it to a list, but not necessarily to the end.
f) **Traverse** a list means traveling through the list.
g) in a **circular** list, the last node has a pointer to the first node
h) In a **doubly-linked** list, each node has a pointer to the one before it and the one after it.

[17] True or False

1. programmer must know in advance how many nodes be needed in a linked list. **(False)**
2. It is not necessary for each node in a linked list to have a self-referential pointer **(true)**
3. In physical memory, the nodes in a linked list may be scattered around **(true)**
4. When the head pointer points to NULL, it signifies an empty list **(true)**
5. Deleting node in linked list is simple, using delete operator to free node's memory. **(False)**

[18] what is the difference between Static and Dynamic data structures

Static = fixed size, implemented as arrays.

Dynamic = can grow in size, need not to specify its size in advance, implemented as linked lists.

[19] What is the difference between appending and inserting a node to a list?
Which is easier to code: appending or inserting?
Discuss the algorithm that used to insert node within a given list?

Append a node means adding it to the end of a list.
Insert a node means adding it to a list, but not necessarily to the end.

The easier to code: **appending**

Inserting a node in the middle of a list. Assuming the nodes in the list are in order.
Create a new node.
Store data In the new node.
If there are no nodes In the list
    Make the new node the first node.
Else
Find the first node whose value is > or = the new value, or the end of the list (whichever is first).
Insert the new node before the found node, or at the end of the list if no node was found.
End If.