Ex No: 1 **Implementation of List ADT**

A List is a sequence of zero or more elements of a given data type, usually represented by $a_1, a_2, a_3, \ldots, a_n$, where $n \geq 0$.

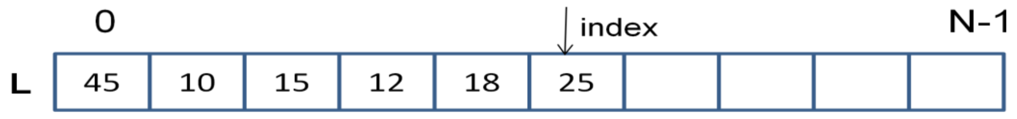
Common operations on List ADT are:

insert(List, item)	- inserts an item into the list*(at first position, anywhere, at last position)*
find (List, key)	- searches an item in the list
delete(List, pos)	- deletes an item from the list, given a position
delete(List, key)	- deletes an item from the list, given an item
first(List)	- returns the first item in the list
last(List)	- returns the last item in the list
prev(List)	- returns the previous item in the list
next(List)	- returns the next item in the list
isEmpty(List)	- returns true if the list is empty
isFull(List)	- returns true if the list is full

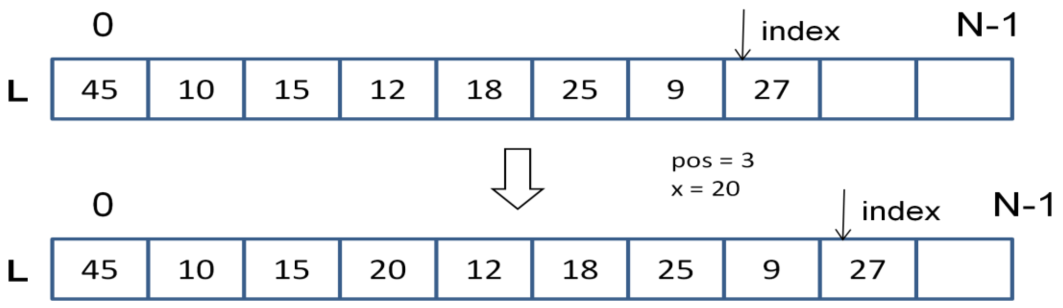
Aim: To create a list of numbers using insert, find and delete List ADT

Procedure:

(i) Array is used for implementation, **L** is the name of list, **N** is the max size of list, **index** tracks the last position of items in the list.

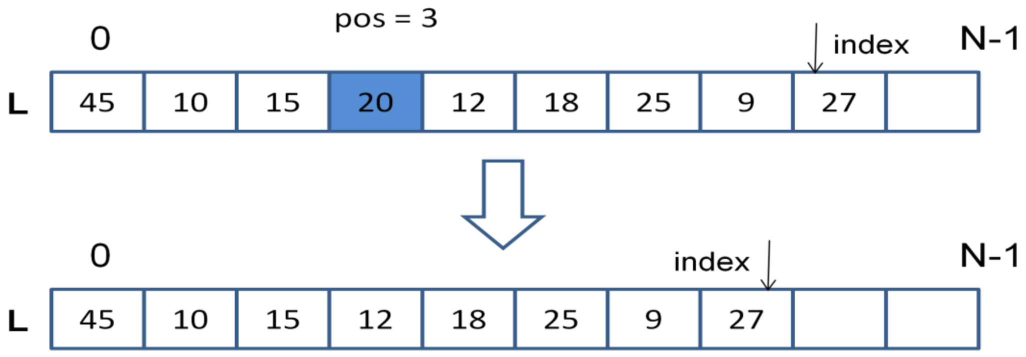


```
insertAnywhere(L, index, pos, x)        // pos tells where to insert, x tells what to insert
begin
   increment index by 1;
   loop
     move elements right such that  [pos] to [pos+1], ..... [index-1] to [index]
   end loop
   L[pos] = x;
   return L;
end
```



```
find(L, x)                                   // x tells what to search
begin
  for i=0 to index
    if L[i] equals x
          print x is found in ith position
          break;
    else
          continue;
   end for
  if (i is greater than index)
     print “x is not found”;
end
```

```
deleteAnywhere(L, index, pos)        // pos tells where to delete
begin
   x = L[pos];
   loop
     move elements left such that  [pos+1] to [pos], ..... [index] to [index-1]
   end loop
   decrement index by 1;
   return x;
end
```



(ii) Structure is used for implementation, where the array is one of the items in it, **L** is the name of list, **N** is the max size of list, **index** tracks the last position of items in the list.

Pseudo code:

```
#define N 10
struct list
{
   int item[N];
   int index;
} L;
```

```
insert(struct list *l, int x)
{
   l->index++;
   l->item[l->index] = x;
}
```

```
int main()
{
    …
  scanf("%d",&x);
// item to be inserted
insert(&L, x);
}
```

Result: Thus the list of odd numbers was created using arrays in C with different List ADT operations.

Sample Questions, *but not limited to***:**(Use arrays or array of structures for implementation)

- Create a list of TamilNadu tourism places. Write suitable functions such that the program insert a new place, searches a given place and deletes any place from the list.

- Create a list of Indian States and their capitals. Write suitable functions such that the program insert a new state/capital, searches a given state/capital by its capital/state and deletes any place from the list.

- Create a list of student details like RegNo, Name, marks of 5 subjects. Write suitable functions such that the program inserts a new student details, searches any student data, and deletes any student data from the list. Calculate total marks of each student and find the class average.

- Create a list of employee details like EmpNo, Name, salary *(basic, HRA, IncomeTax)*. Write suitable functions such that the program inserts a new employee details, searches any employee data, and deletes any employee data from the list. Calculate gross and net salary for each employee.

Assessment Rubrics for Ex 1:

Parameters	**Max Marks**	**Actual Marks**
Uniqueness of the problem selection	10	
Diagrammatic Representation	10	
Code & Output - 1 (i) Arrays 1 (ii) Array of Structures	10	
Viva (Online Quiz)	10	
Adherence to the template for documentation (Record)	10	
Total	50	

Staff Signature	