Array Implementation of list:

Array: A set of data elements of same data type is called array. Array is a static data structure i.e., the memory should be allocated in advance and the size is fixed. This will waste the memory space when used space is less than the allocated space. An array implementation allows the following operations.

The basic operations are:

a. Creation of a List.

b. Insertion of a data in the List

c. Deletion of a data from the List

d. Searching of a data in the list

Global Declaration:

```
int list[25], index=-1;
```

Note: The initial value of index is -1.

Create Operation:

Procedure

- The list is initially created with a set of elements.
- Get the no. of elements (n) to be added in the list.
- Check n is less than or equal to maximum size. If yes, add the elements to the list.
- Otherwise, give an error message

Program

```
void create()

{

int n,i;
printf("\nEnter the no.of elements to be added in the list");

s.canf("%d",&n); if(n<=maxsize) for(i=0;i<n;i++)

{

scanf("%d",&list[i]); index++;

}

else

printf("\nThe size is limited. You cannot add data into the list");

}
```

Insert Operation:

Procedure:

- Get the data element to be inserted.
- Get the position at which element is to be inserted.
- If index is less than or equal to maxsize, then Make that position empty by altering the position of the elements in the list.
- Insert the element in the poistion.
- Otherwise, it implies that the list is empty.

Program

```
void insert()

{

int i,data,pos;
printf("\nEnter the data to be inserted");
scanf("%d",&data);
printf("\nEnter the position at which element to be inserted");
scanf("%d",&pos);
if(index<maxsize)
{
for(i=index;i>=pos-1;i--)
list[i+1]=list[i];
index++;
list[pos-1]=data;
}
else
printf("\nThe list is full");
}
```

Deletion

Procedure

- Get the position of the element to be deleted.
- Alter the position of the elements by performing an assignment operation, list[i-1]=list[i], where i value ranges

  from position to the last index of the array.

Program:

```
void del()

{

int i,pos;

printf("\nEnter the position of the data to be deleted");

scanf("%d",&pos);

printf("\nThe data deleted is %d",list[pos-1]);

for(i=pos;i<=index;i++)

list[i-1]=list[i];

index--;

}
```

Display

Procedure

- Formulate a loop, where i value ranges from 0 to index (index denotes the index of the last element in the array.
- Display each element in the array.

Program

```
void display()

{
```

```
int i; for(i=0;i<=index;i++) printf("\t%d",list[i]);

}
```

Limitation of array implementation

- An array size is fixed at the start of execution and can store only the limited number of elements.
- Insertion and deletion of array are expensive. Since insertion is performed by pushing the entire array one position down and deletion is performed by shifting the entire array one position up.