S10_2

1D-Arrays

Objectives

To learn and appreciate the following concepts:

- Declare, initialize and access 1D array.

- Write programs using common data structures namely arrays and strings and solve problems.

Session outcome

- At the end of session student will be able to

  → Declare, initialize and access 1D array

  → Write programs using 1D array

Arrays – recap

1D Array:

-    Syntax: type array_name[size];

- Memory Requirement:
Total size =size *(sizeof(data_type));

- Initialization:
type array-name [size]={list of values}

- Write and Read:
for(i=0;i<n;i++)   for(i=0;i<n;i++)
 scanf("%d",&a[i]);   prinft("%d\n",a[i]);

WAP to insert an element to an array at a given position

int a[100], n,i, pos, ele;

 scanf("%d",&n); // number of elements

printf("\nEnter the elements of array:");

for(i=0;i<n;i++)

```
 scanf("%d",&a[i]);
```

printf("\nEnter the element and position of insertion:");

```
 scanf("%d %d",&ele,&pos);

for(i=n; i>=pos; i--)  //shift the elements to right

     a[i]=a[i-1];

a[pos-1] = ele;  //ele is inserted at  the specified  pos.

n = n + 1;     // increment the count of no of elements
```

printf("\nThe array after insertion is:");

for(i=0;i<n; i++)   `printf("%d\n",a[i]);`

> **Example : insert 9 at 2nd position**
> **a[ ]={1, 2, 3, 4, 5}**

> **New array after inserting 9 :**
> **a[ ]={1, 9, 2, 3, 4, 5}**

 MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

WAP to delete an element from an array

printf("enter no of numbers");

```
scanf("%d",&n);
```

printf("enter n numbers \n");

for(i=0;i<n;i++)

```
    scanf("%d",&a[i]);
```

printf("enter the position at which the element to be deleted");

```
scanf("%d",&pos);
```

```
for(i=pos-1; i<n-1; i++)
```

```
    a[i] =a[i+1];
```
 //shift the elements to left

```
n = n-1;
```
     //decrement the count of no of elements

for(i=0;i<n;i++)

```
    printf("%d",a[i]);
```

> **Example : delete ele at 2ⁿᵈ position**
> **a[ ]={1, 2, 3, 4, 5}**

> **New array after deleting 2:**
> **a[ ]={1, 3, 4, 5}**

Insert an element into a sorted array

Read array elements (in sorted order) & element '**ele**' to be inserted

```
//finding position
for(i=0;i<n;i++)

        if (ele<a[i])

            break;
 pos = i+1;  //position of insertion

for(i=n; i>=pos; i--)  //shift the elements to right

  a[i]=a[i-1];

a[pos-1] = ele;  //ele is inserted at  the specified  pos.

n = n + 1;      // increment the count of no of elements
```
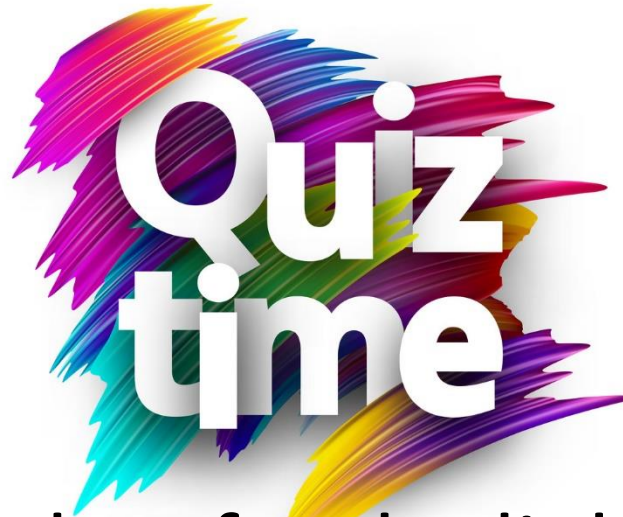
Example: insert 3 into the array
a[ ] = {1, 2, 4, 5, 6}

New array after inserting 3 :
a[ ] = {1, 2, 3, 4, 5, 6}

Go to posts/chat box for the link to the question

submit your solution in next 2 minutes

The session will resume in 3 minutes

Tutorials on Array

- Write a C program to find average of an 1-D array.

- Write a C program to find second largest element in an array.

- Write a C program to find union and intersection of two arrays.

Largest and second largest element in an array

/* assume first element of the array as the largest & second largest */

```
largest1 = array[0];
largest2 = array[1];

    for (i = 1; i < MAX; i++)
    {
        if (array[i] >= largest1)
        {
            largest2 = largest1;
            largest1 = array[i];
        }
        else if (array[i] > largest2)
        {
            largest2 = array[i];
        }
    }
```

Example: array[ ] = {22, 44, 34, 9, 21}

```
44 is largest
34 is second largest
```

Summary

- Problems on 1D arrays