V. Program to Cal hypotenuse (2)

Page (1)

Data Structure and Algo

Array:- Collection of similar data element, that deny with be of similar data type (int, float etc).

- Array are stored in Consequtive memory loc.
- type name[size]; {type can be either int float etc}

// Accessing the element of an array

```
//set each element of the array to -1
int i, Marks[10];
for (i=0; i<10; i++)
    ~~Marks = []~~
    Marks[i] = -1
```

there is no single element statement that can read, access, or print all the element of an array. we used to use loop for this.

// Calculating the Address of Array Element

[Address of data element, A[K] = BA (A) + W(K - lowerbd)]

Array ← ~~Add~~ A[K] ↳ index

BA → Base Address, W = size of one element in try

Example Base address = 1000

int marks[] = {99, 67, 78, 98, 90, 92, 58, 78}

Mark[4] = 1000+2(4-0) = 1000+8 = 1008

1000	1002	1004	1006					
99	67	78	98	90	92	58	78	-1
0	1	2	3	4	5	6	7	

// Assigning Value to individual Elements :-

// Cal the length of an Array

[length = upper bound - lower bound + 1]

Example Marks[5]

Marks[0] = 68, Marks[1] = 78, Marks[2] = 98, Marks[3] = 25

Marks[4] = 67

	68	78	98	25	67
Marks[] = {	0	1	2	3	4 }

Length = 4 - 0 + 1 = 5 ⇒ Length = 5

// Sorting values in Arrays

- When we declear an Array, we are just allocating space in memory, none value is Assigned to Array.
- There are three ways to store values in an Array

1. initialize value during deleration

type a[5] = {90, 87, 10, 20, 10}
int.

2. input value from the keyboard

3. Assign value to individual.

2.

```
int i, marks[10];
for (i=0; i<10; i++)
    scanf("%d", &marks[i]);
```

// Assigning Values to individual Elements :-

```
int i, arr1[10], arr2[10];
  arr[10] = { 0,1,2,3,4,5,6,7,8,9 };
    for (i=0; i<10; i++)
        arr2[i] = arr[i];
```

Code to copy array at the individual level.

```
int i, arr[10];
  for (i=0; i<10; i++)
      arr[i] = i*2;
```

Code with even numbers.

Operation on Array

- Traversing An Array
- Inserting an element in An Array
- Searching an element in An Array
- Del an element from an Array
- Merging two array
- Sorting an array in ascending (or) des

Traversing An Array

↳ accessing each & every element of array by some purpose

Algorithm

```
Step 1: [INITIALIZATION] SET I = lower-bound
Step 2: Repeat Steps 3 to 4 while I <= upper-bound
                                   ↳ executes the last elem
Step 3:     Apply process to A[I]
Step 4:         SET I = I+1 // increment.
        [END OF LOOP]
Step 5: EXIT.
```

Programming. Q: Write a program to read and display n numbers using an array

```
#include<stdio.h>
int main()
{   int i, n, arr[50];
// Enter the no of elements
      Scanf("%d", &n);
      for (i=0; i<n; i++)
       {  Printf("\n arr[%d] = " i);
           Scanf ("%d" , & arr[i]);
        }
      Printf(" \n the elements are ");
       for ( i=0 ; i<n ; i++)
            Printf ( " \t %d" arr[i] );
       return 0;
       }
```

Q. Write a program to find the mean of n nos. using array

```
      int Main()
           {  int i, n, arr[20], Sum=0;
                 float mean = 0.0 ;
              Scanf(" %d", &n);
              for ( i=0 ; i<n ; i++)
              {  Print (" \n arr[%d] = ", i);
                    Scanf(" %d" , arr[i]);

                 }
```

```
for (i=0; i<n; i++)
        Sum = Sum + arr[i];
    Mean =(float) sum/n;
  Printf (" \n The Sum of the array elements = %d", Sum);
  Printf(" \n the mean of array = %.2f", mean);
  Return 0;

  }.
```

Q: Write a program to print the position of the smallest number of n numbers using arrays:-

```
#include <stdio.h>
   int main()
 { int i, n, arr[20], Small, Pos;
 // Enter the no. of element in the array
   Scanf ("%d" &n);
    for ( i=0; i<n ; i++)
          Scanf ("%d" &arr[i] );

Imp. [ Small = a[0];
       Pos = 0;
        for ( i=1 ; i<n; i++)
          {
              { if (arr[i] < Small)
              {   { Small = arr[i];
              {           Pos = i;
          }         }
   Pf ("\n The Smallest element is : %d", Small);
   Pf (" \n The Pos of Smallest element %d", Pos);
Return 0; }
```

Q. ~~Thire~~ 2nd largest of n number using Array. (6)

```
int main()
 { int i, n, arr[20], large, second_large;
     Scanf(" %d", &n);
     for( i=0; i<n; i++)
      Scanf(" %d", &arr[i]);
     large = arr[0]
     for( i=1 ; i<n; i++)
      {
            if (arr[i] < large)
          large = arr[i];
      }
     Second_large = arr[1]
      for( i=0; i<n ; i++)
        {      if (arr[i] != large)

                 { if (arr[i] > second_large)

                         Second_large = arr[i];
                 }
        }
  printf(" \n no You entered are:");
   for( i=0; i<n; i++)
           printf(" \t %d", arr[i]);
   pf(" \n the largest of these no. is : %d" large);
   pf (" \n the Sec larges of these no. %d " Second_lar);
 return 0; }.
```

Q. Write a program to enter n number of digit. form a Number using these digits.

```
#include<stdio.h>
#include<math.h>
        int main()
            { int  number=0, digit[10], i, noofdigit;
         Scanf("%d", &noofdigit);
           for( i=0; i< noofdigit ; i++)
              {  print("Enter the digit of position %d" i+1);
                   Scanf ("%d", &digit[i]);
              }
           i=0 ;
           While ( i< noofdigit)
              {  number = number + digit[i] * Pow(10,i)
                        i++;
              }
           printf("\n The Number is : %d" number);
           return 0;
        }
```

input 4   Enter the digit of position 1 : 2

```
                                    :   : 3
                                    :   : 0
                                    :   : 9
```

The Num is 9032

⑧ Writ a proggoam to find dublicate value

```
#include< stdio.h>
int main()
{
 int array[10]i,n,j,flag=0;
 Scanf ("%d", &n);
 for (i=0 ; i<n ; i++)
   { pf (" \n array[%d] = " i);
      Scf(" %d " &array[i];
   }
  for (i=0; i<n ; i++)
   { for (j=i+1 ; j<n ;j++)
      {
         if (array[i]== arr[j] && i!=j)
        { flag=1;
     printf ("\n Dublicate no found at loc %d and
                        %d" ,i,j);
      }
      }
   }
 if (flag==0)
     Pruf ("\n No Dublicate found");
 return 0;
}
```

Output

```
Enter the size of array:
     5
array[0]=1
array[1]=2
array[2]=3
array[3]=2
array[4]=5
Dublicate no fount at
  1 and 3
```