ISM
We inspire you to learn

ARRAYS

Array Requirement ???

- Problem : we wants to read 20 numbers, process them and print them out.
  - Conventional Way:
    - declare 20 variables like number0, number1, …, number19.
    - Read one by one all the numbers
    - Process them
    - Print one by one all the numbers
  - Using Arrays:
    - Declare an array which have 20 elements
    - Using loop, which iterate 20 times, read element value.
    - Process them
    - Using loop, which iterate 20 times, print element value.

What is an Array?

- An array is a fixed-size, sequenced collection of elements of the same data type.
- As an array is sequenced collection, we can refer to the elements using subscripts. The subscripts indicate the ordinal number of the element counting from the beginning of the array.
  - In mathematics, for an array of n numbers, we may refer them as:

    numbers_0, numbers_1, numbers_2, …, numbers_{n-1}

Declaration of Arrays

- Specify the identifier type and the name of identifier
- Specify the size of the array within array operators [ ]
  - Size is total numbers of elements in the array
  - This is an integer
  - A variable cannot be used to declare the size of the array. It must be a constant expression or defined constant (memory constant is not allowed).
  - This is also known as dimension and declaration is known as dimensioning.

Example: declare array named num with 20 elements of integer type

```
int num[20];
```

Accessing Elements in Arrays

- Use an index to access individual elements in an array.
  - Index must be an integral value or an expression that evaluates to an integral value.
  - Subscript or Index starts from 0 (not 1). This means, an array declared to have 10 elements, can have index values of 0..9.
    - Array's name is symbolic reference (constant pointer) for the address to the first byte of the array.

An entire array can not be assigned to another array

Array Initialization

- Basic Initialization

  int num [5] = { 5, 7, 15, 24, 99};

- Partial Initialization

  int num [5] = { 3, 7};

- Initialization to all zeroes

  int num[1000] = {0};

- Initialization without size

  int num [ ] = {5,7,15,18,11,10};

Note: while initialization of an 1D array, size is optional.

2-D ARRAY

ISM
We inspire you to learn

DECLARATION OF 2-D ARRAY

```
int matrix[4][4];
```

INITIALIZATION OF 2-D ARRAY

```
int matrix[][4]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16};
```

OR

```
int matrix[][4]={{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}};
```

Note: while initialization of an 2D array Row size is optional, but Column size is mandatory.

2D Sample program

programs\arrays\2darr.c

DECLARATION OF MULTI-DIMENSIONAL ARRAY

```
type array_name[s1][s2][s3]…[sn];
```

```
int a[3][4][5];
```

STRINGS

C [Level1] : What is Strings

•Group of characters is called string.
•Many languages treats group of characters are strings.
•A string is a one-Dimensional array of characters terminated by NULL (‘\0’)

Ex:- char name[]={‘I’,’s’,’m’,’\0’};

•Each char occupies one bytes.
•Char array elements are stored in contiguous memory allocations.
•‘\0’ denotes the end of the string.
•Many functions are working by using this concept.
•Another initialization is

Char name[]=“ism”;

•Here ‘\0’ is not necessary C inserts automatically.

C [Level1] : Basic concepts in Strings

ISM
We inspire you to learn

Char array only ended with ‘\0’.

No other arrays are ended with ‘\0’.

Example 1:

```
{
        char name[]=“ism”;
        int i=0;
        while(i<3)
        {
                printf(“%c”,name[i]);
                i++;
        }
}
```

C [Level1] : Basic concepts in Strings

Example 2:

```
{
        char name[]=“ism”;
        int i=0;
        while(name[i]!=‘\0’)
        {
                printf(“%c”,name[i]);
                i++;
        }
}
```

Example 3:

```
{
        char name[]=“ism”;
        printf(“%s”,name);
}
```

Example 4:

```
{
        char name[20];
        printf(“\n enter string::”);
        scanf(“%s”,name);
}
```

C [Level1] : Basic concepts in Strings

- •String should not exceed the dimension of the character array.
- •Scanf() is not having the capability of receiving multi-word string.
- •Solution for this problem is

```
        gets(name);
        Scanf(“%[^\n]s”,name);
```

Arrays and Functions

- Passing individual element is similar to any other variable.
- We can also pass the whole array to function.
  - C does not pass the values to function
    - To save lot of memory and time
  - Instead of passing the whole array, C passes the address of the array.

- Passing the whole array:
  - The function must be called by passing only the name of the array.
  - In the function definition, the formal parameter must be an array type; the size of the array does not need to be specified.

Example – Passing Array

..\..\student reference (Level-1)\programs\arrays\arrfun2.c

Passing 2D Array to function

programs\arrays\arrfun3.c