UNIT- III

ARRAYS: Collection of elements of same datatype in sequence

fixed size

* Since Array is a sequenced collection of similar or related data items, it provides a convenient structure for representing data ie one of the data structures in C.

* Array is a derived data type

Types of Arrays:

1. One dimensional Array int a[5]
2. Two dimensional Array int a[3][3]
3. Three dimensional Array int a[2][3][3]
4. Character Array or Strings char name[10]

Multidimensional

Array Representation:

90	97	96	93	95	91	100	92	98	99	94
0	1	2	3	4	5	6	7	8	9	10 → Array Index

→ Marks of 11 students

size of array - 11 (Array length)

First index - 0

Last index - 10

[Since starts from 0, last index is (size is - One)]

Declaration of array:

datatype arrayname[size];

Eg int a[10];

→ a is an array of integer type with the size of 10.

Initializing an array:

* by assigning initial values at declaration
* getting values from user or assigning values using for loop

* By assigning initial values:

double balance[5] = {1000.0, 2.0, 3.4, 7.0, 50.0};

- No. of elements in the braces cannot be larger than 5

if declared, double balance[] = {1000.0, 2.0, 3.4, 7.0, 50.0};

→ size of array is not mentioned
→ array will be created just big enough to hold the initial values

balance[4] = 50.0;

→ 5th element in the array will have value 50.0

double salary = balance[9];

- assigns 10th element of array balance to the variable salary

* getting input from user or direct assign
- Using for loop

```
#include <stdio.h>

int main()
{
    int n[10]; // n is array of 10 integers
    int i,j;

    /* Initialize elements of array */
    for(i=0; i<10; i++)
    {
        n[i] = i+100;
    }
    /* Output the values */
    for(j=0; j<10; j++)
    { printf(" Element [%d] = %d\n", j, n[j]);
    }
    return 0;
}
```

```
for(i=0; i<10; i++)
{
    printf(" Enter the value of array Element
                n[%d]\n", i);
    scanf(" %d ", &n[i]);
}
```

Find average of n numbers using array

```
#include <stdio.h>
Int main()
{
    int marks[10], i, n, sum = 0, average;
    printf(" Enter n:");
    scanf("%d", &n);
    for(i = 0; i<n; i++)
    {
        Printf("\nEnter number %d:", i+1);
        Scanf(" %d" &marks[i]);
        sum += marks[i];
    }
    average = sum/n;
    printf(" Average = %d", average)
    return 0;
}
```

Output:

```
Enter n: 5
Enter number 1: 45
Enter number 2: 35
Enter number 3: 38
Enter number 4: 31
Enter number 5: 49
Average = 39
```

Two-Dimensional Array:

Declaration: datatype arrayname[row size][column size]

Example:

```
#define ROWS 5
#define COLUMNS 5
void main()
{
    int row, column, product[ROWS][COLUMNS];
    int i, j;
    printf("Multiplication Table\n\n");
    for(j=1; j<=COLUMNS; j++)
        printf("%4d", j);
    printf("\n");
    printf("----------\n");
    for(i=0; i<ROWS; i++)
    {
        row = i+1;
        printf("%2d|", row);
        for(j=1; j<=COLUMNS; j++)
        {
            column = j;
            product[i][j] = row * column;
            printf("%4d", product[i][j]);
        }
        printf("\n");
    }
}
```

O/P Multiplication Table

	1	2	3	4	5
1	1	2	3	4	5
2	2	4	6	8	10
3	3	6	9	12	15
4	4	8	12	16	20
5	5	10	15	20	25

Character Array:

* A string is a sequence of characters that are treated as a single data item.
* Group of characters defined between double quotation marks is a string constant

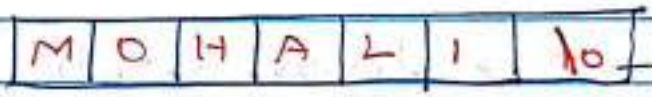
```
printf (" Good Morning ");
↳ o/p: Good Morning
```

Declaration:

```
char stringname [size];
Eg char city [7];        (Array size 7)
```

Initialization:

```
char city [7] = " MOHALI ";
        (or)
char city[7] = {'M', 'O', 'H', 'A', 'L', 'I',
```



* Array variable stores address of first index location

To read strings:

```
char city[7];
scanf(" %s", city);
```

Example:

```
void main()
{
    char word1[10], word2[20];
    printf (" Enter the text ");
    scanf (" %s", word1);
```

```
    scanf("%s", word2);
    printf("Word1 = %s word2 = %s",
                word1, word2);
}
```

It is also possible to specify:

```
scanf("%.ws", name);
```

↓ width of the word

* The width w is equal to or greater than the number of characters typed in.
* width w is less than the number of characters in the string. The excess characters will be truncated and left unread.

STRING FUNCTIONS: (Using <string.h> header file)

strlen(s) → finds length of string s
strlwr(s) → converts string s to lowercase
strupr(s) → converts s to uppercase
strcat(s1,s2) → combines/appends s2 to end of s1
strncat(s1,s2,n) → First n characters of s2 is concatenated at end of s1
~~strcpy~~
strcpy(s1,s2) → copies s2 to s1
strcmp(s1,s2) → compares s1 & s2
strncmp(s1,s2,n) → returns positive or negative by comparing first n characters of s1 & s2
strcmpi(s1,s2) → compares s1 & s2 ignoring case.

strnicmp(s1, s2, n) → compares first n characters of s1 & s2 ignoring cases

strdup(s) → duplicates string s

strchr(s, c) → finds first occurrence of character c in string s

strrchr(s, c) → finds last occurrence of character c in string s

strstr(s1, s2) → finds first occurance of s2 in s1

strset(s, c) → sets all the characters in the string to c

~~strnset~~

strrev(s) → reverses the string

Typecast functions:

atof() → Converts string to float
atoi() → converts string to int
atol() → converts string to long
itoa() → converts int to string
ltoa() → converts long to string

Example:

```
#include <stdio.h>
#include <string.h>
void main()
{   char Mystring[10];
    printf(" Enter a text which
           has less than 10 chara
```

```
    gets(Mystring);
    puts("\nThe entered string is:\n");
    puts(Mystring);
}
```

OUTPUT:

Enter a text which has less than 10 chars.
HELLO
The entered string is:
HELLO

Example:

```
#include <stdio.h>
#include <string.h>
void main()
{
    char s1[10], s2[20], s3[30];
    int x, l1, l2, l3;
    printf("\n Enter two string constants\n");
    scanf("%s %s", s1, s2);
    // compare the strings
    x = strcmp(s1, s2);
    if(x != 0)
    {
        printf("\n Strings are not equal");
        strcat(s1, s2);
    }
    else
        printf("\n Strings are equal");
    // Copy s1 to s3
    strcpy(s3, s1);
    // Finding length of strings
```

```
l1 = strlen(s1);
l2 = strlen(s2);
l3 = strlen(s3);
printf("\n\nS1 = %s \t length = %d characters",
                                    s1, l1);
printf("\n\nS2 = %s \t length = %d characters",
                                    s2, l2);
printf("\n\nS3 = %s \t length = %d characters",
                                    s3, l3);
}
```

Output:

```
Enter two string Constants
NEW DELHI
Strings are not equal
S1 = NEW DELHI    length = 8 characters
S2 = DELHI        length = 5 characters
S3 = NEW DELHI    length = 8 characters
```

Output 2:

```
Enter two string constants
PUNJAB PUNJAB
Strings are equal
S1 = PUNJAB    length = 6 Characters
S2 = PUNJAB    length = 6 Characters
S3 = PUNJAB    length = 6 Characters
```