Arrays

An ordinary variable can store single value at one time. If we want to store marks of 100 different student, then there are two options:-

(i) Construct 100 different variables to store the marks of the 100 different students.

(ii) Construct one variable which is capable of storing or holding all the hundred values.

(Array

Subscripted variable

So, array is a collection of homogeneous (same type) of elements like all integer, all character or all real type. Set of similar datatype stored in a contiguous memory location is called array.

Declaration of an array

int marks[10]; int n[i];

Here int is the type of the data stored in the array, marks is the array name and subscripted variable, 10 is the subscript or dimension of the array

n is subscripted variable.
i is subscript.

Initialization of array :-

int x[5] = { 5, 10, 50, 90, 80 };

The array index start with (0) and end with (subscript -1). So, the array element will be initialize as :
x[0] = 5, x[1] = 10, x[2] = 50, x[3] = 90, x[4] = 80.

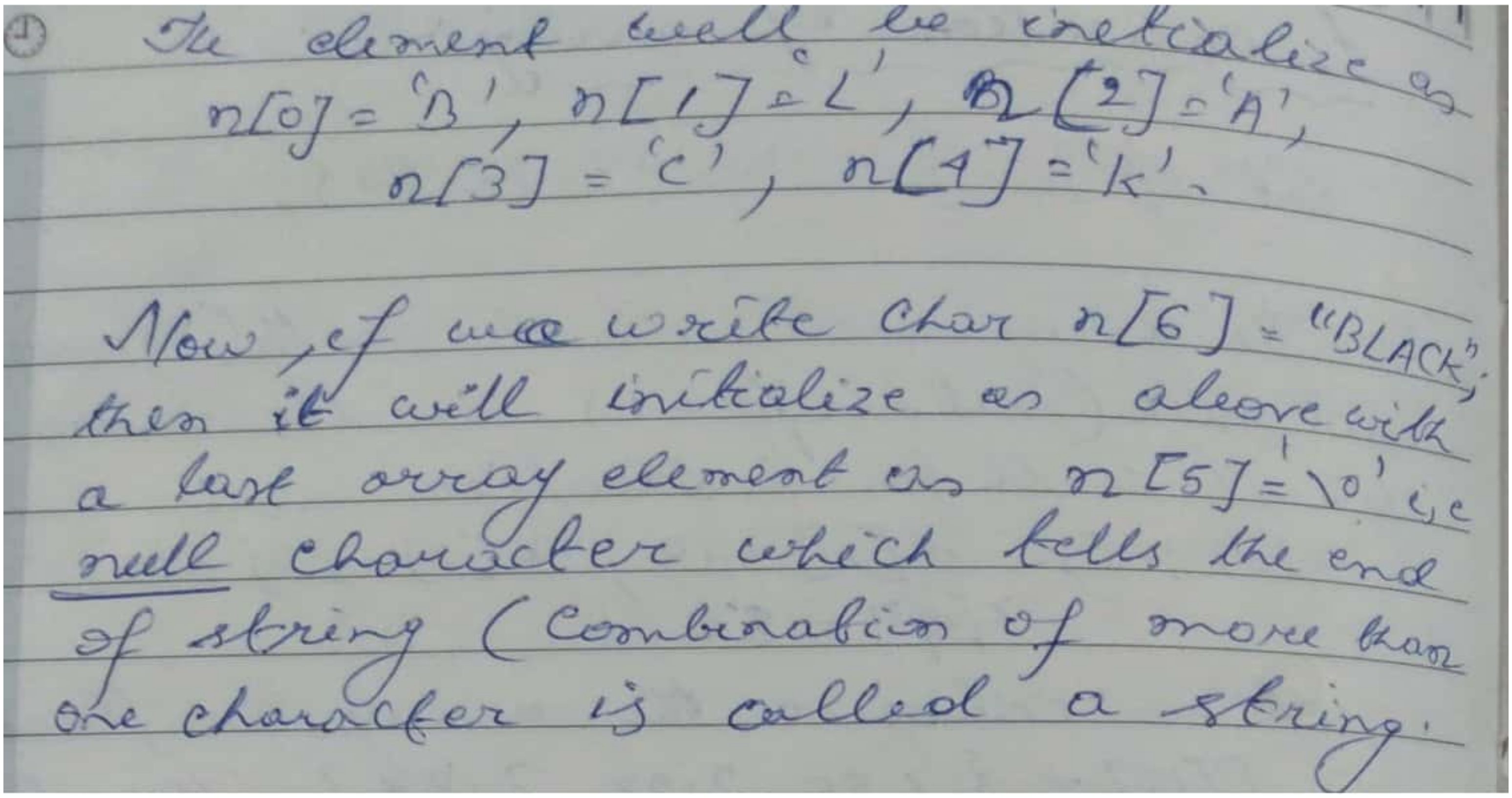
If we initialize the array as float P[10] = { 1.55, 2.97, 3.87 }, then the element will be saved as :
P[0] = 1.55, P[1] = 2.97, P[2] = 3.87 and rest of the array will be initialize to zero i;e P[3], – – – = P[9] = 0.0 ;

23 August • Tuesday

We can also initialize the array as :
int x[] = { 5, 10, 50, 90, 80 }

It will also have the same initialization of the above.
In case of character array, we can initialize in two ways :-

Char n [5] = { 'B', 'L', 'A', 'C', 'K' }
or
Char n [5] = { "BLACK" }



The element will be initialize as
n[0] = 'B', n[1] = 'L', n[2] = 'A',
n[3] = 'C', n[4] = 'K'.

Now, if we write char n[6] = "BLACK";
then it will initialize as above with a last array element as n[5] = '\0' i.e null character which tells the end of string (Combination of more than one character is called a string.

Scanned with CamScanner

A simple Array program:-

```
main()
{
  float avg, sum = 0;
  int i;
  int marks[30];
  for(i=0; i<=29; i++)
     {
        printf("Enter Marks");
        scanf("%d", &marks[i]);
     }
  for(i=0; i<=29; i++)
     Sum = sum+(marks[i]);

  avg = sum/30;
  printf("n Average marks=%f",
                              avg);
}
```

More on Arrays

Array is a very popular data type with C programmers. This is because of the convenience with which arrays lend themselves to programming. The features which make arrays so convenient to program would be discussed below, along with the possible pitfalls in using them.

Array Initialization

So far we have used arrays that did not have any values in them to begin with. We managed to store values in them during program execution. Let us now see how to initialize an array while declaring it. Following are a few examples that demonstrate this:

```
int num[ 6 ] = { 2, 4, 12, 5, 45, 5 } ;
int n[ ] = { 2, 4, 12, 5, 45, 5 } ;
float press[ ] = { 12.3, 34.2, -23.4, -11.3 } ;
```

Note the following points carefully:

(a) Till the array elements are not given any specific values, they are supposed to contain garbage values.

(b) If the array is initialised where it is declared, mentioning the dimension of the array is optional as in the 2nd and 3rd examples above.

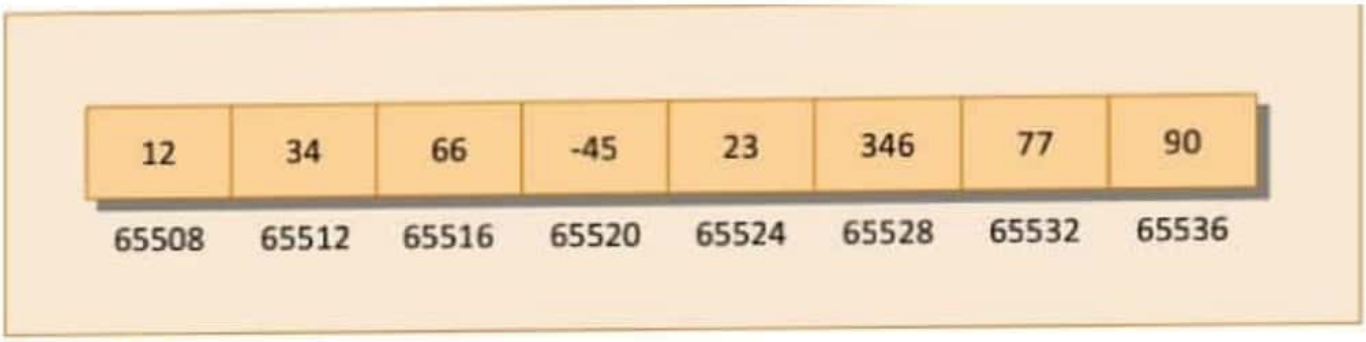


Figure 13.1

Bounds Checking

In C, there is no check to see if the subscript used for an array exceeds the size of the array. Data entered with a subscript exceeding the array size will simply be placed in memory outside the array; probably on top of other data, or on the program itself. This will lead to unpredictable results, to say the least, and there will be no error message to warn you that you are going beyond the array size. In some cases, the computer may just hang. Thus, the following program may turn out to be suicidal:

```
# include <stdio.h>
int main( )
{
    int  num[ 40 ], i ;

    for ( i = 0 ; i <= 100 ; i++ )
        num[ i ] = i ;
    return 0 ;
}
```

Thus, to see to it that we do not reach beyond the array size, is entirely

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

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int i=0, n, arr[20];
    clrscr();
    printf("\n Enter the number of elements:");
    scanf("%d", &n);
    for(i=0;i<n;i++)
    {
        printf("\n Arr[%d] = ", i);
        scanf("%d", &arr[i]);
    }
    printf("\n The array elements are ");
    for(i=0;i<n;i++)

        printf("Arr[%d] = %d\t", i, arr[i]);
    return 0;
}
```

Output

```
Enter the number of elements: 5
Arr[0] = 1
Arr[1] = 2
Arr[3] = 3
Arr[4] = 4
Arr[5] = 5
The array elements are
Arr[0] = 1    Arr[1] = 2    Arr[3] = 3
Arr[4] = 4    Arr[5] = 5
```