Date
Page

Unit - 2

Array:- An array is defined as an ordered set of similar data items. All the data items of an array are stored in consecutive memory locations in RAM. The elements of an array are of same data type and each item can be accessed using the same name.

Declaration of an array:- We know that all the variables are declared before they are used in the program. Similarly, an array must be declared before it is used. During declaration, the size of the array has to be specified. The size used during declaration of the array informs the compiler to allocate and reserve the specified memory locations.

Syntax:- data_type array_name[n];

where, n is the number of data items (or) index (or) dimension.

0 to n-1 is the range of array.

Ex:- int a[5];

float x[10];

Initialization of Arrays:- The different types of initializing array:

1. At Compile time – (i) Initializing all specified memory locations.
(ii) Partial array initialization.
(iii) Initialization without size.
(iv) String initialization.
2. At Run time.

1. Compile Time Initialization – We can initialize the elements of an array in the same way as the ordinary variables when they are declared. The general form of initialization of an array is

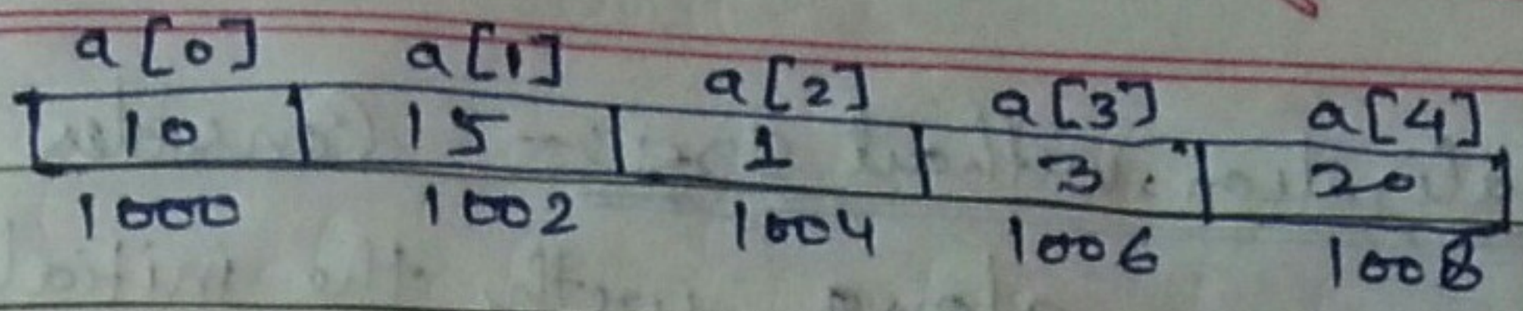
type array_name [size] = { list of values };

(i) Initializing all specified memory locations :- An Array can be initialized at the time of declaration when their initial values are known in advance. An array elements can be initialized with data items of type int, char etc.

Ex:- int a[5] = { 10, 15, 1, 3, 20 };

During compilation, 5 contiguous memory location are reserved by the compiler for the variable a and all these locations are initialized as show in fig.

Suppost first location of initial element of an array is 1000.



Ex:-

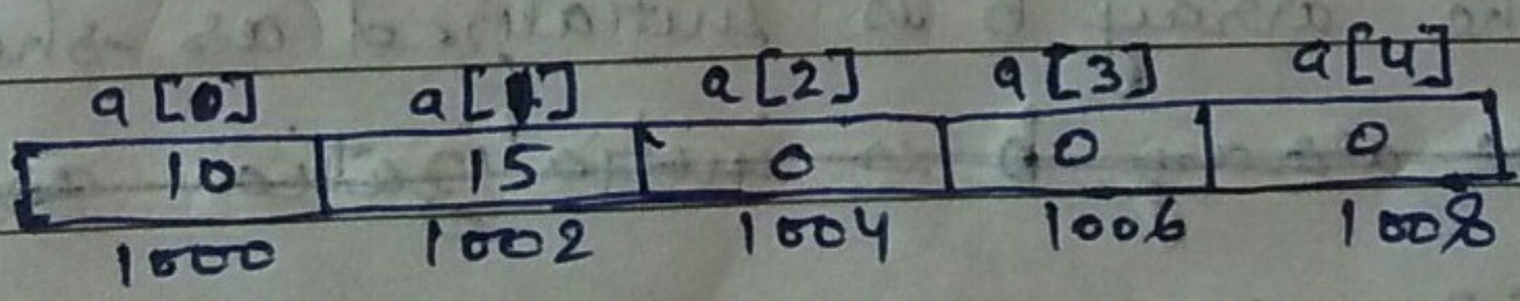
int a[3] = {9,2,4,5,6}; // error

number of initial values are more than the size of an array.

ii) Partial array initialization:- Partial array initialization is possible in C language. If the number of values to be initialized is less than the size of the array, then the elements will be initialized to zero automatically.

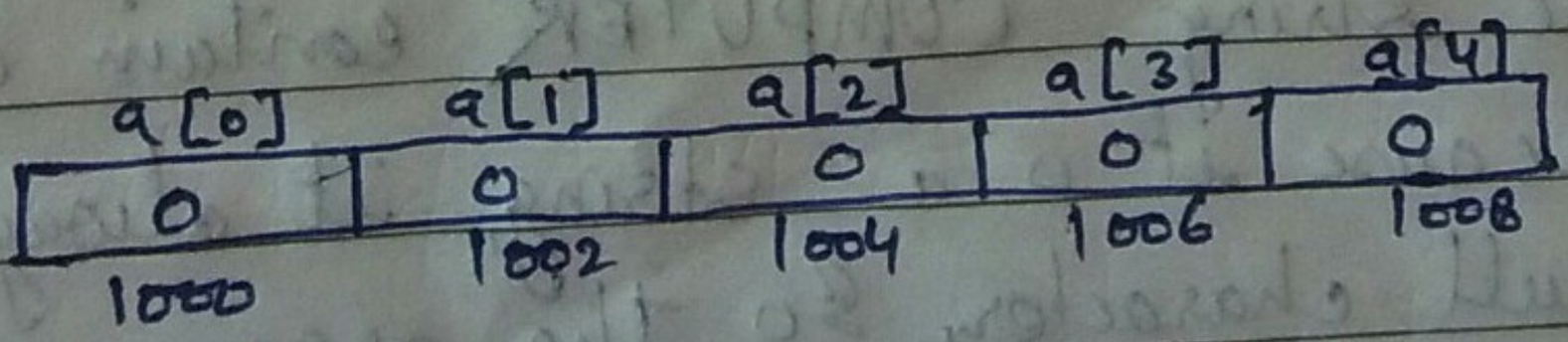
Ex:- int a[5] = {10, 15};

Even though compiler allocates 5 memory locations. The compiler initializes first two locations with 10 and 15, the next set of memory locations are automatically initialized to 0's by compiler as shown in figure.



Initialization with all zeroes:-

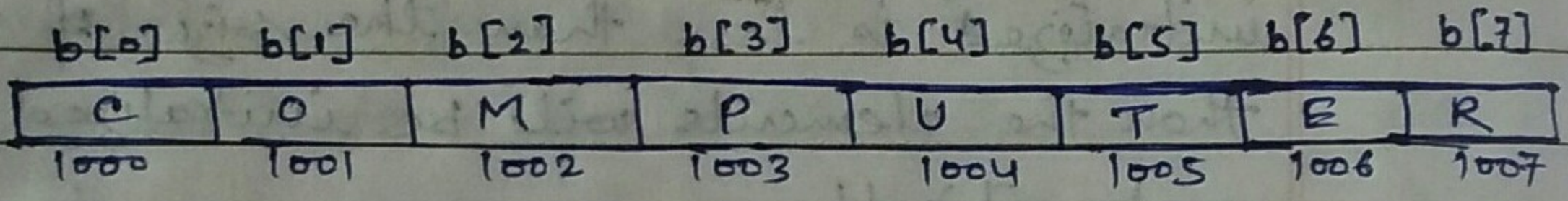
Ex:- int a[5] = {0};



(iii) Initialization without size:- Consider the declaration along with the initialization.

Ex:- char b[] = {'C','O','M','P','U','T','E','R'};

In this declaration, we have not specified exact number of elements to be used in array b, the array size will be set of the total number of initial values specified. So the array size will be set to 8 automatically. The array b is initialized as shown in figure.

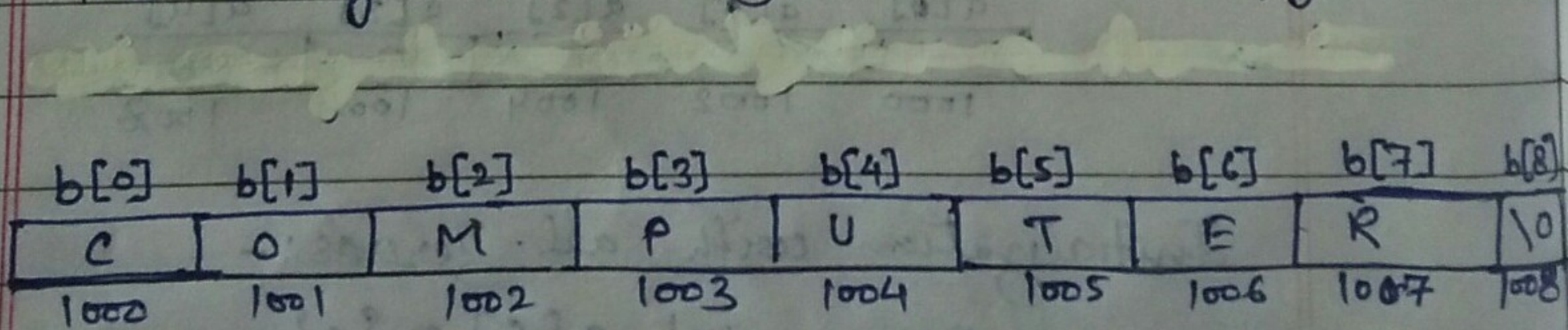


Ex:- int ch[] = {1, 0, 3, 5} // array size is 4.

(iv) Array initialization with a string:- Consider the declaration with string initialization.

Ex:- char b[] = "COMPUTER";

The array b is initialized as shown in figure.



the string "COMPUTER" contain 8 characters, because it is a string, it always ends with null character. So the array size is 9.

Ex:- char b[9] = "COMPUTER" // Correct
char b[8] = "COMPUTER" // Wrong

2. Run Time Initialization:- An array can be explicitly initialized at run time. This approach is usually applied for initializing large arrays.

Ex:- scanf can be used to initialize an array.

```
int x[3];
scanf("%d%d%d", &x[0], &x[1], &x[2]);
```

The above statements will initialize array elements with the values entered through the Key board. or