C programming

Trainer : Nisha Dingare

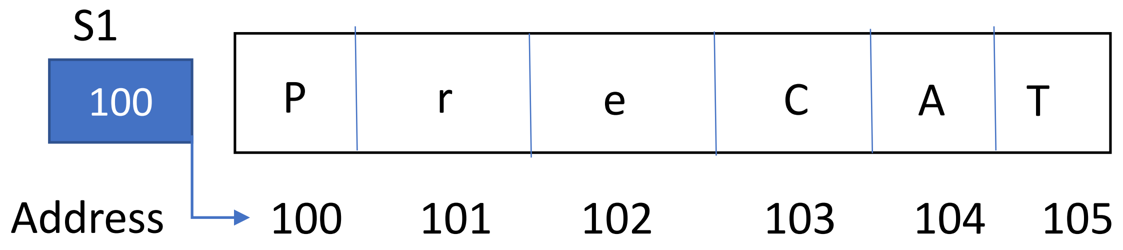
Email : nisha.dingare@sunbeaminfo.com

Strings and character arrays :

- Character Array :

  Collection of character elements.

  char s1[5] = {‘P’,’r’,’e’,’C’,’A’,’T’};



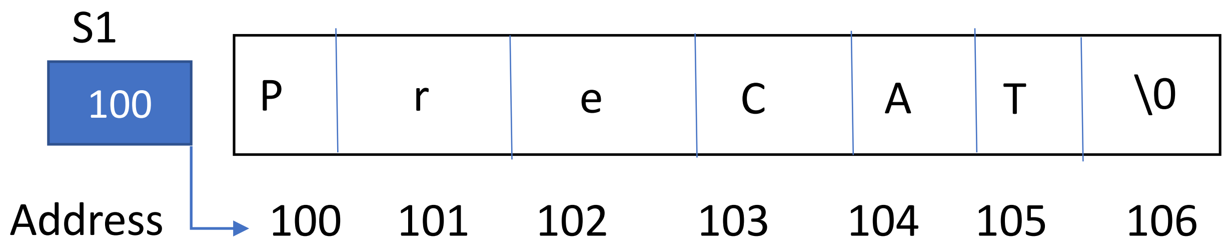
- String:

  Collection of character elements with sentinel element ‘\0’

  char s1[] = {‘P’,’r’,’e’,’C’,’A’,’T’,’\0’};

  char s1[] = “PreCAT”;

  char *s1 = “PreCAT”;



- Size :

  Always need to reserve 1 byte extra for sentinel element NULL.

String :

- Not a primitive datatype.
- C compiler provides special library function to handle strings.
- These library functions are declared in string.h
- e.g.
- strlen
- strcpy
- strcmp
- strcat
- strstr
- strupr
- strlwr
- strrev
- strchr

String :

- Example :

```
char arr[5] = “abcde”;
 int j;
 for(j=0; j<5; j++)
             printf(“%c”,arr[j]);
```

- Accepting string as a input

```
char str[20];
scanf(“%s”,str); // input
printf(“%s”,str); // output

char str[20];
gets(str);
puts(str);
```

String scan sets :

- %s
- %[^\n]s //scan upto \n (single line)
- %[^.]s // scan upto . (multiple line)
- %[0-9]s// scan upto digits
- %[^0-9]s// scan upto alphabets
- %[A-Z]s// scan upto capital
- %[^a-z]s// scan upto capital
- %[^A-Z]s// scan upto small letter
- %[a-z]s// scan upto small letter

2-D Arrays :

- Arrays that we have considered till now are one dimensional arrays, a single line of elements.
- Often data comes naturally in the form of a table, e.g., spreadsheet, which need a two-dimensional array.
- Two-dimensional (2D) arrays are indexed by two subscripts, one for the row and one for the column.
- Example: int a[2][3];

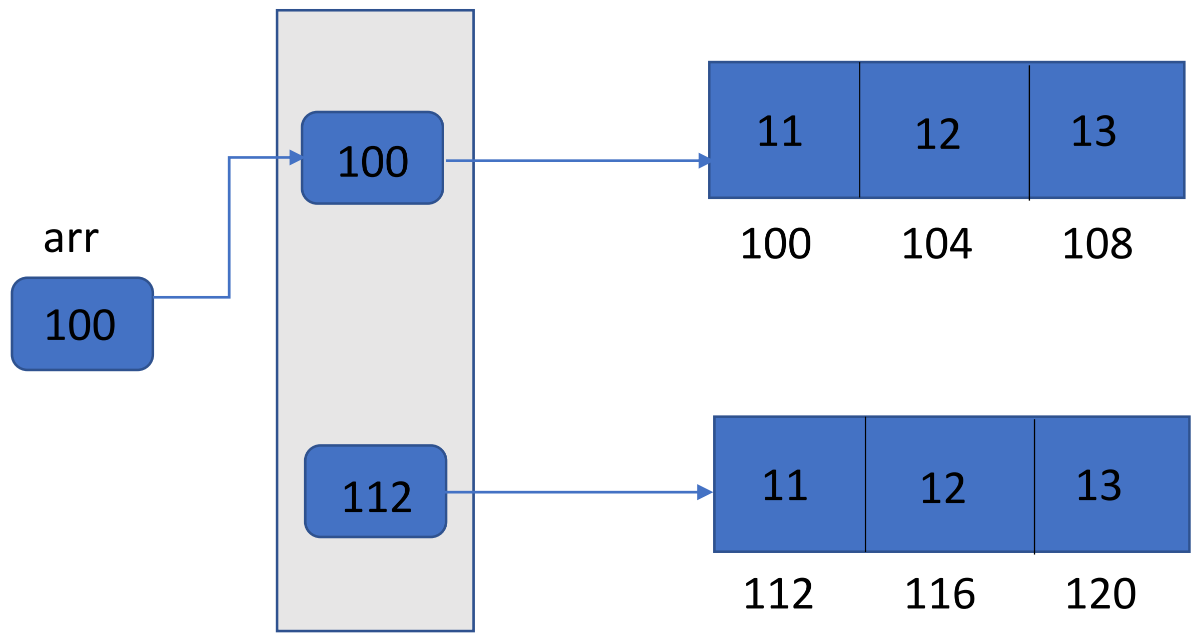
Logically it may be viewed as a two-dimensional collection of data, two rows and three columns each location is of type int.

```
int arr[2][3] = {{11,22,33},{44,55}};

int arr[2][3] = {11,22,33,44,55};
```

Multi Dimensional Array :

- int arr[2][3] = {{11,22,33},{44,55,66}};



arr[1][1]	**==**	***(*(arr+1)+1)**
arr	**=**	**100**
Address of row / pointer to int		
arr+1	**=**	**112**
Address of row / pointer to int		
***(arr+1)**	**=**	**112**
Address of int		
***(arr+1)+1**	**=**	**116**
Address of int		
***(*(arr+1)+1)**	**=**	**55**

2D array Declarations :

- **Valid Declarations :**
- 1.int mat[2][2]={{1,1},{1,2},{2,1},{2,2}}; //allowed
- 2.int mat1[ROW][COL]={{1,1},{1,2},{2,1},{2,2}}; //allowed
- 3.int mat3[][COL]={{1,1},{1,2},{2,1},{2,2}}; // allowed
- 4.int mat4[2][2];

- **Invalid Declarations :**
- 1.int mat[][]={{1,1},{1,2},{2,1},{2,2}};// not allowed
- 2.int mat2[ROW][]={{1,1},{1,2},{2,1},{2,2}}; //not allowed

Thank You!!