



C PROGRAMING

by Akshita Chanchlani @ Sunbeam Infotech



Day3 : 1D Array and String



1D Array



Array

- Array is collection of similar data elements in contiguous memory locations.
- Elements of array share the same name i.e. name of the array.
- Array are identified by unique index in the array called as subscript.
- Array index starts from 0 to n-1 (where n is size of an array).
- Checking array bounds should be done by programmer.
- Size of array is fixed (it cannot be grow/shrink at runtime).

```
int main() {  
    int i, arr[5] = {11, 22, 33, 44, 55};  
    for(i=0; i<5; i++)  
        printf("%d\n", arr[i]);  
    return 0;  
}
```

	0	1	2	3	4
arr	11	22	33	44	55
	400	404	408	412	416
	arr[0]	arr[1]	arr[2]	arr[3]	arr[4]



Points to be noted

- If array is initialized partially at the time of declaration then elements which are not initialized are set with default value 0(ZERO).
- It is necessary to specify last dimension in case of all dimension of array only in case of 1D array we can skip last dimension providing members of array are initialized at the time of declaration.
- When we request array a location is identified with array name stores always first element address of first element because of which array interchangeability with pointer is possible
- The array name is treated as address of 0th element in any runtime expression.
- Pointer to array is pointer to 0th element of the array.



Passing Array to Function

- Arrays are passed to function by address.
- The address of starting element of array (Base address) is passed to the function.
- The address is collected in a pointer.
- We can use pointer as well as array notation to access array elements in the function.
- In a function definition, a formal parameter that is declared as an array is actually a pointer.
- When an array is passed, its base address is passed call-by-value.
- The array elements themselves are not copied.
- As a notational convenience, the compiler allows array bracket notation to be used in declaring pointers as parameters.



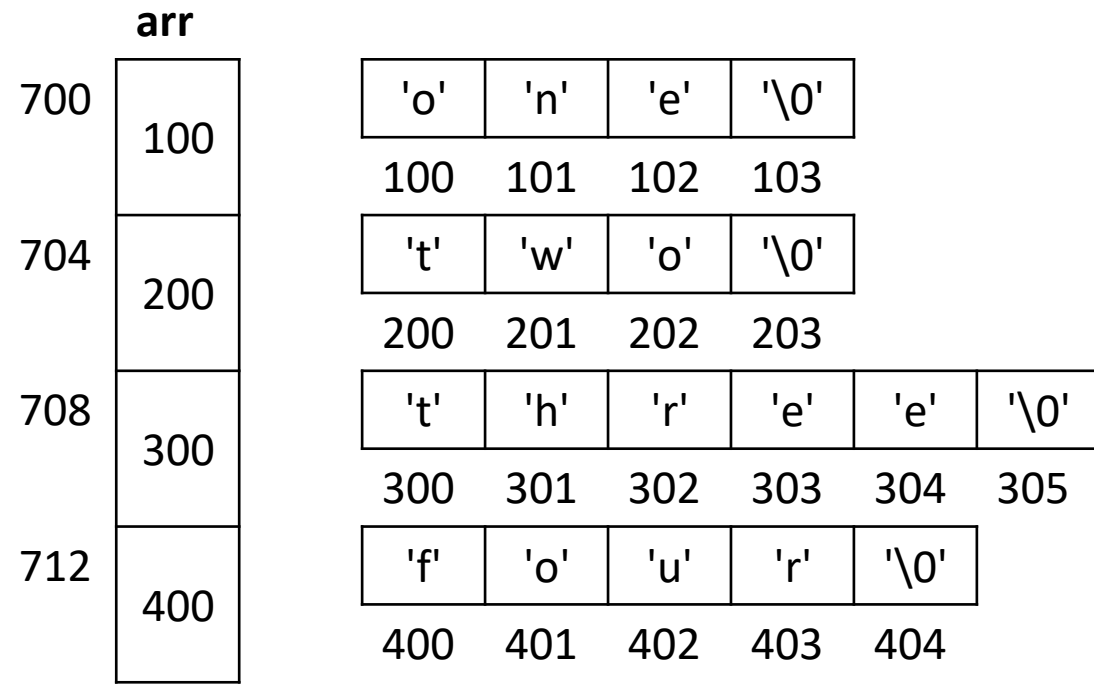
Array and Pointer

- An array name represents the address of the beginning of the array in memory.
- When an array is declared, the compiler must allocate a base address and a sufficient amount of storage (RAM) to contain all the elements of the array.
- The base address of the array is the initial location in memory where the array is stored.
- It is the address of the first element (index 0) of the array.
- Example:
 - `int arr[5] ;`
 - `arr[i]` is internally resolved as `*(arr+i)`
 - `arr[i] == *(arr+i)`
 - `*(i+arr) == i[arr]`
 - `int *ptr=arr;`
 - `*(ptr+i) == ptr[i]`
 - `*(i+ptr) == i[ptr]`



Array of pointers

```
char *arr[] = { "one", "two", "three", "four" };  
for(i = 0; i < 4; i++)  
    puts(arr[i]);
```



String



String

- Character array vs String
 - Character array: It is collection of characters
 - String: Array of characters where always last element is NULL.

- Example :

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

- String input/output

- char str[20];
- scanf("%s",str); /*Input*/
- printf("%s",str); /*Output*/
- gets(str); /*Input*/
- puts(str); /*Output*/
- scanf("%[^\n]", str); // scan whole line



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



String functions

- C library have many string functions.
- They are declared in string.h
 - `strlen()` – `size_t strlen(const char *s);`
 - `strcpy()` – `char* strcpy(char *dest, const char *src);`
 - `strcat()` – `char* strcat(char *dest, const char *src);`
 - `strcmp()` – `int strcmp(const char *s1, const char *s2);`
 - `strchr()` – `char* strchr(const char *s, int ch);`
 - `strstr()` – `char* strstr(const char *s1, const char *s2);`
 - `strrev()` – `char* strrev(char *s);`





Thank you!

Akshita Chanchlani <akshita.chanchlani@sunbeaminfo.com>

