# String in C

# String

- Most common use of one dimensional array is string
- C has no built in string datatype
- One dimensional character array terminated by a null (`'\0'`)
- `'\0'` & `'0'` are not same
  - Value of `'\0'` is 0 and considered `false`
  - Value of `'0'` is 48
- Array size must be at least one byte larger than the string size to make room for the null

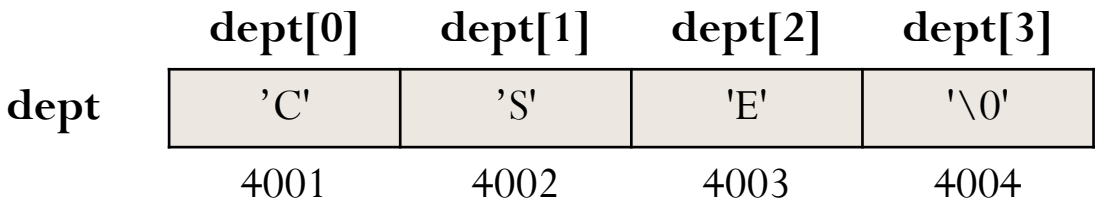
| 'C' | 'S' | 'I' | '1' | '2' | '1' | '\0' |
|---|---|---|---|---|---|---|

String Size: 6, Array Size: 7

- Terminating null is important
  - Indicates where string ends
- A string constant is automatically null-terminated by the compiler

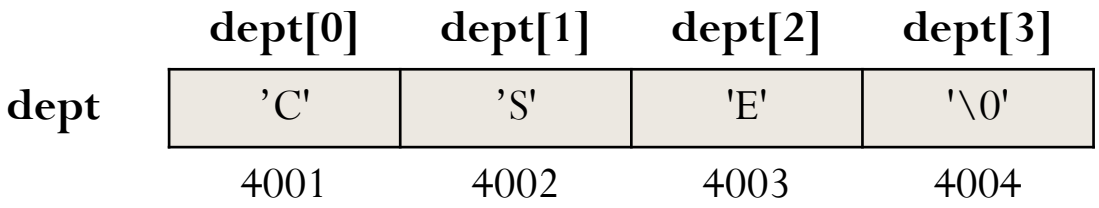
# String

- char dept[]={'C', 'S', 'E', '\0'};



# String

- char dept[]=''CSE";
  - Shortcut for initializing string
  - '\0' is not necessary in this declaration



# String

```
#include<stdio.h>

int main()
{
        char course[]="CSI121";
        int i=0;
        while(course[i])
        {
                printf("%c\n", course[i]);
                i++;
        }
        return 0;
}
```

Output:

```
C
S
I
1
2
1
```

# String Read using `scanf()`

- %s is used in scanf
- Reads characters untill `ENTER` pressed
- `ENTER` key is not stored, replaced with null character
- No bound checking
- Can not read multi word string
  - "Department Name: CSE"
- scanf("%s", s);
  - Just the array name is used
  - No & (ampersand) is used

# String Read using `gets()`

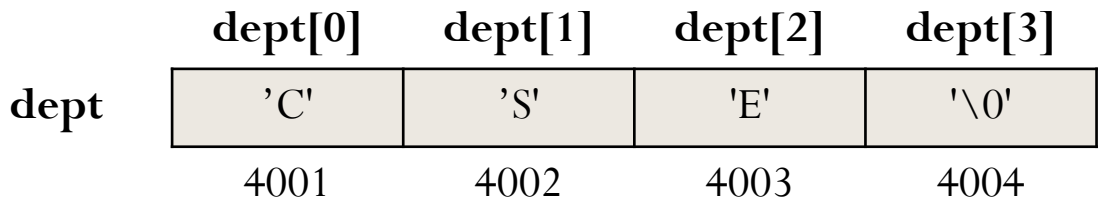
- gets()
  - Library function
  - Defined in stdio.h
  - Call it using the name of the character array without using any index or ampersand (&)
    - gets(s)
  - Reads characters untill `ENTER` pressed
  - `ENTER` key is not stored, replaced with null character
  - No bound checking
  - Can receive multiword string

# String Write

- Using %s in printf
  - printf("%s", s);
- Using puts()
  - puts("hello")
  - puts(s)

# String

- Each character occupies one byte of memory



# String Library Functions

- strlen : Finds the length of the string
- strcat : Appends one string at the end of the other
- strcpy( to, from): Copies one string into another
- strncpy : Copies first n characters of one string into another
- strcmp (s1, s2): Compares two strings
  - Returns 0 if same
  - -ve if s1 less than s2
  - +ve if s1 greater than s2
- strchr : Finds first occurrence of a given character in a string

# String Library Functions

- strcpy

```
char dest[]="";
char src[]="CSI121";
for(int i=0; src[i]!='\0',;i++)
    dest[i]=src[i];
dest[i]='\0';
```