

# C, C++, DSA in depth

## Strings



Saurabh Shukla (MySirG)

## Agenda

- ① String Introduction
- ② null character
- ③ User input
- ④ String functions
- ⑤ Function call by passing string
- ⑥ Handling multiple strings.

## String

- String is a sequence of characters, terminated at null character.
- Strings are handled in char arrays.

```
char str[10];
```

## Initializing char array during declaration

0	1	2	3	4	5	6	7	8	9
B	H	O	P	A	L				

```
int main()
```

```
{
```

```
char str[10] = { 'B', 'H', 'O', 'P', 'A', 'L' };
```

```
int i;
```

```
for ( i=0 ; i<=9 ; i++ )
```

```
    printf("%c", str[i]);
```

```
return 0;
```

i  
[ 6 ]

```
}
```

BHOPAL

# ASCII

characters    codes    screen

'\0'	0	Blank space
' '	32	
'G'	48	
'g'	57	
'@'	64	
'A'	65	A
'B'	66	B
'Z'	90	
'a'	97	

# Printing String

0 1 2 3 4 5

B H O P A L O O O O

```
int main()
```

```
{
```

```
char str[10] = { 'B', 'H', 'O', 'P', 'A', 'L' };
```

```
int i;
```

```
for(i=0; str[i] ; i++)
```

```
printf("%c", str[i]);
```

```
return 0;
```

```
}
```

'A' '\0'  
65 0

i str[i]  
0 B  
1 H  
2 O  
3 P  
4 A  
5 L  
6 \0

## Improve your code

```
int main()
{
    char str[10] = {'B', 'H', 'O', 'P', 'A', 'L'};
    int i;
    for (i=0; str[i]; i++)
        printf("%c", str[i]);
}
```

## Use of null character

```
int main()
```

```
{
```

```
    char str[10] = { 'B', 'H', 'O', 'P', 'A', 'L' };
```



```
}
```

%S

```
int main()
{
    char str[10] = {'B','H','O','P','A','L'};
    printf("%s", str);
}
```

# Calculating length of the String

```
int main()
```

```
{
```

```
    char str[10] = "BHOPAL";
    int i;
    for(i=0; str[i]; i++);
    printf("Length is %d", i);
    return 0;
```

0 1 2 3 4 5 6  
[BHOPAL 0]

```
}
```

## String Constant

"BHOPAL" ← String Constant  
string Literal

char str[10] = "BHOPAL";

0 1 2 3 4 5 6 7 8 9  
██████████  
B H O P A L 10

~~&str~~

str ✓

~~&str[0]~~ ✓

~~str[]~~

## Scanf()

- scanf uses delimiters to separate input data
- By default delimiters are
  - space
  - tab space
  - new line character

"Saurabh Shukla"  
↑  
space

- scanf is not capable to input multiword string

## Taking input from User

- `scanf` is not capable to input `scanf()` multiword string
- because space, tab, new line characters are delimiters
- we will not use `scanf` for string input

`gets()`

- `gets()` is capable to input multiword string

gets()

only for strings

one string at a time

scanf()

multiple values

MySirG Education

Services Private Limited

↑  
unreliable

fgets( )

fgets( arrayname , inputsize , stdin )

# Memory Concept

```
int i;  
char str[20] = "My Sir G";
```

## option 1

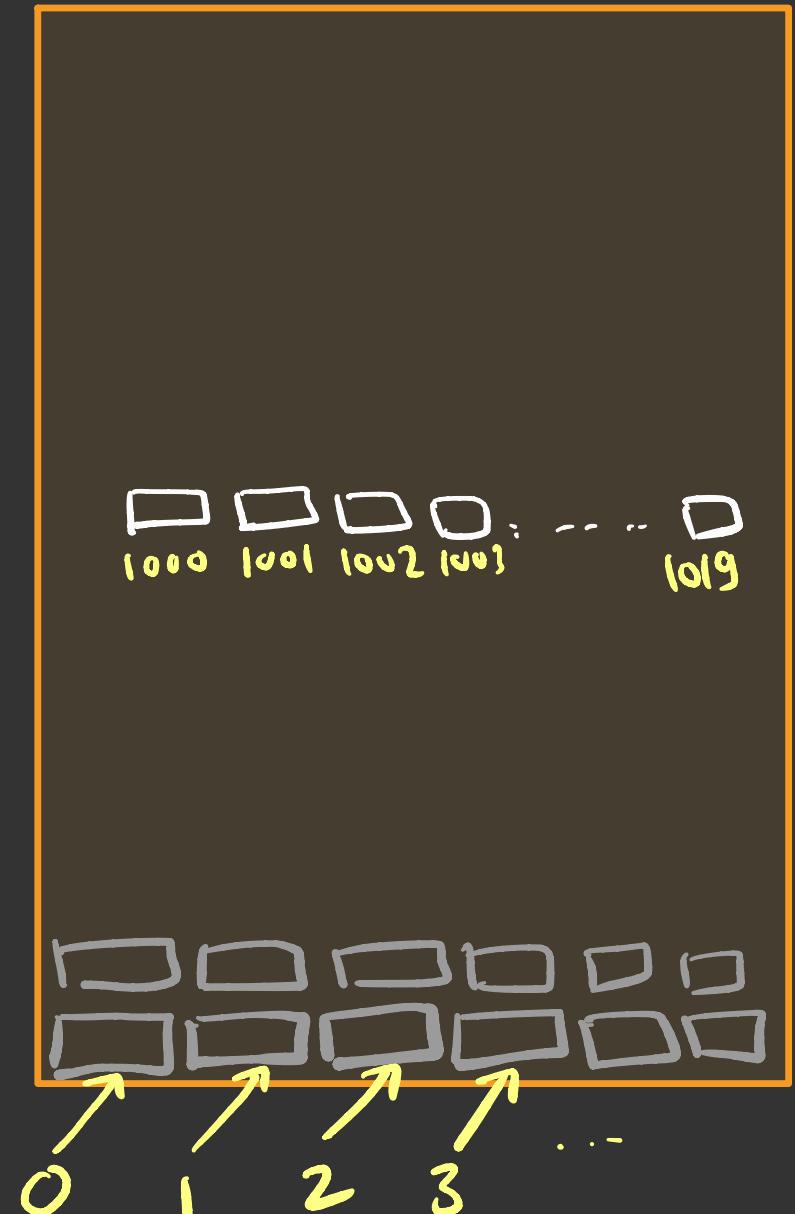
```
for(i=0; str[i]; i++)  
    printf("%c", str[i]);
```

## option 2

```
printf ("%s", str);
```

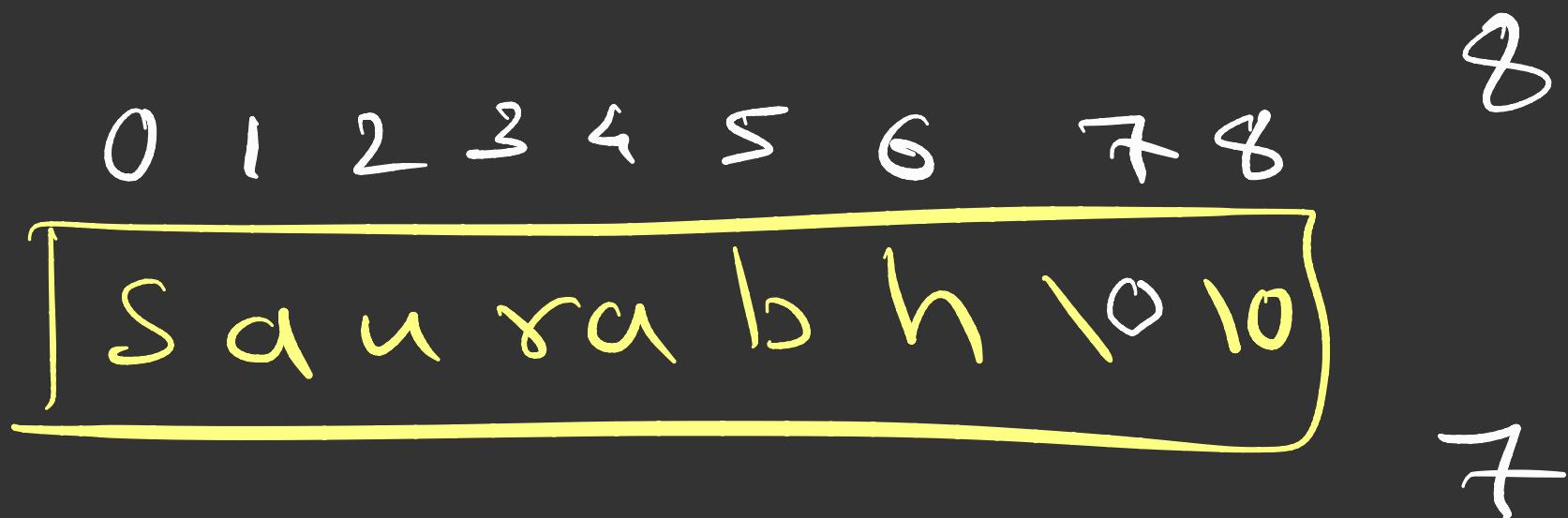
str ≈ 1000 ≈ &str[0]

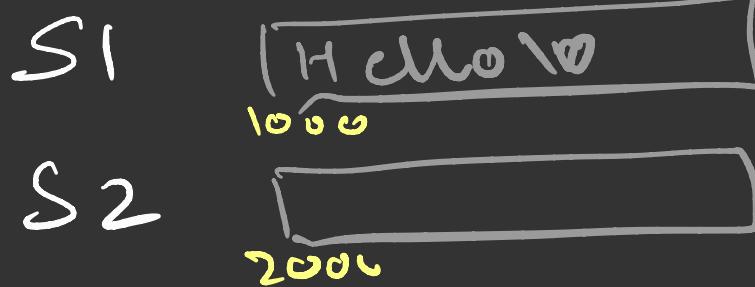
position number of bytes in  
program's memory is called  
an address



## String Functions

```
#include <string.h>
int strlen(char *);    l = strlen(s);
char * strcpy (char *, char *);
char * strcat (char *, char *);
int strcmp (char *, char *);
```





S1 ↳ 1000  
 S2 ↳ 2000

```

char s1[] = "Hello";
char s2[20];
strcpy(s2, s1);
  
```

strcat

S1 → ~~"Ram"~~ "RamRaheem"  
 S2 → "Raheem"

strcat(S1, S2);

strcat(S1, "");

strcat(S1, S2); → "Ram Raheem"

$s1 \rightarrow "Rahul"$   
 $s2 \rightarrow "Rajesh"$

-1  
0  
+1

$\Leftarrow$  strcmp( $s1, s2$ )

- 1  $s1$  comes before  $s2$  in dictionary
- 0 Same
- +1  $s1$  comes after  $s2$  in dictionary

## Function call by passing String

```
fi( "Bhopal"); } void fi(char s[])
fi( str); { }
```

## Handling Multiple Strings

```
char str[3][10];
int i;
printf("Enter three words");
for(i=0; i<=2; i++)
{
    fgets(str[i], 10, stdin);
    str[i][strlen(str[i])-1] = '\0';
}
str[0]
str[1]
str[2]
```

0 1 2 3 4 5 6 7 8 9

0 [ ]

1 [ ]

2 [ ]

char str[3][10] = { "Ravi", "Arjun", "Palak" };

0 

	0	1	2	3	4	5	6	7	8	9
0	R	a	v	i		\	0			

1 

	A	r	j	u	n	\	0
--	---	---	---	---	---	---	---

2 

	P	a	l	a	K	\	0
--	---	---	---	---	---	---	---





