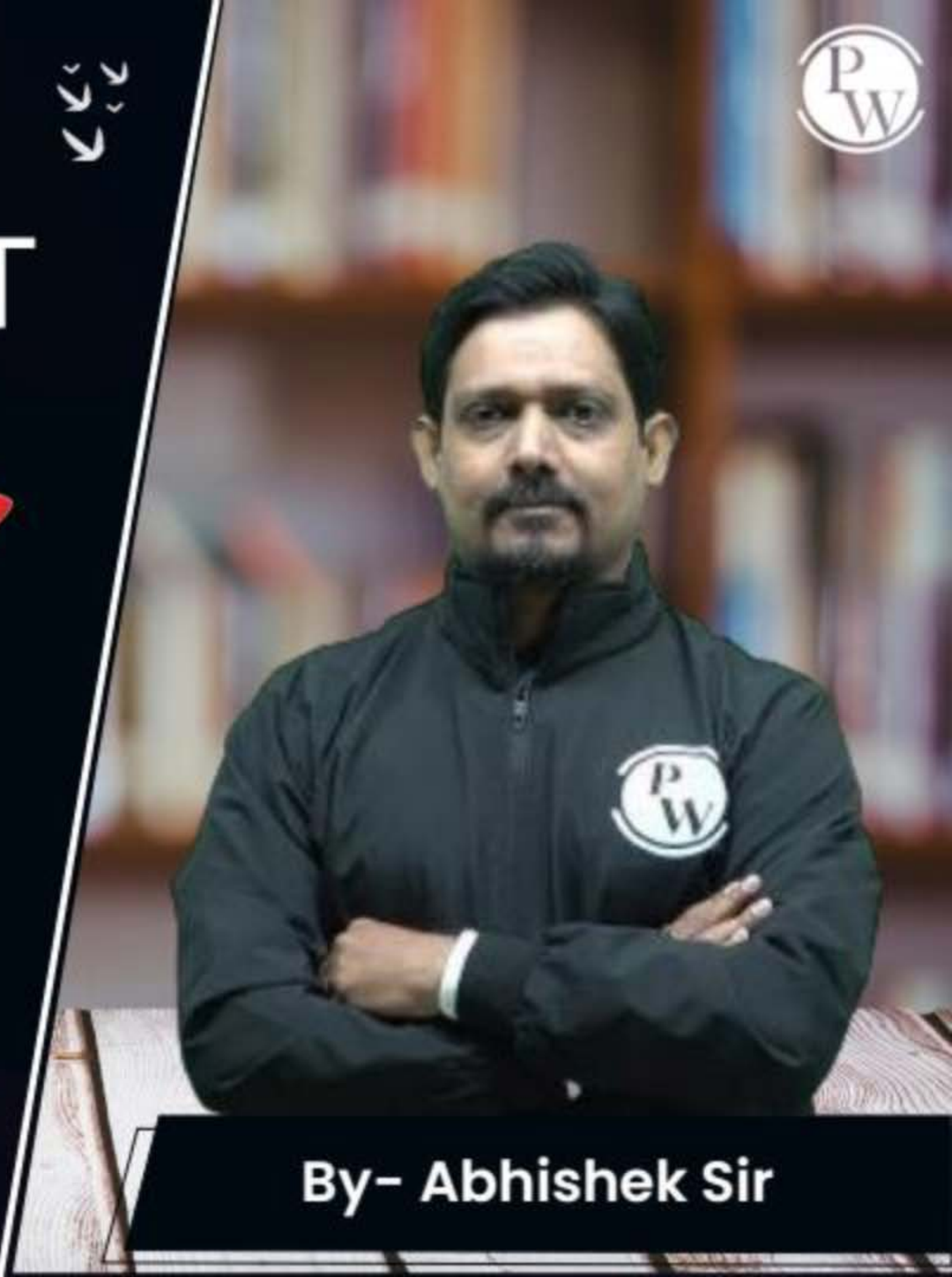
Computer Science & IT

C Programming

String

Lecture No. 01



By- Abhishek Sir

Recap of Previous Lecture

- Topic
- Topic
- Topic
- Topic
- Topic

2D - 1-D array

problem

Slide

Topics to be Covered

- Topic
- Topic
- Topic
- Topic
- Topic

string

array of character

Slide

String

String
Null character
Print string
Two ways to declaration of string
Array of string
How to print array of string
Array of string using character pointer

String

String is array of character

String is token "Hello World"

" "

double quotes

Slide

String

char = 1B

100	101	102	103	104	105	106
s	t	r	i	n	g	\0

```
char ch[] = "string";
```

```
printf("%s", ch); // string
```

```
printf("%s", ch+2); // ring
```

Address first letter

```
ch++; // Error Constant being Modified
```

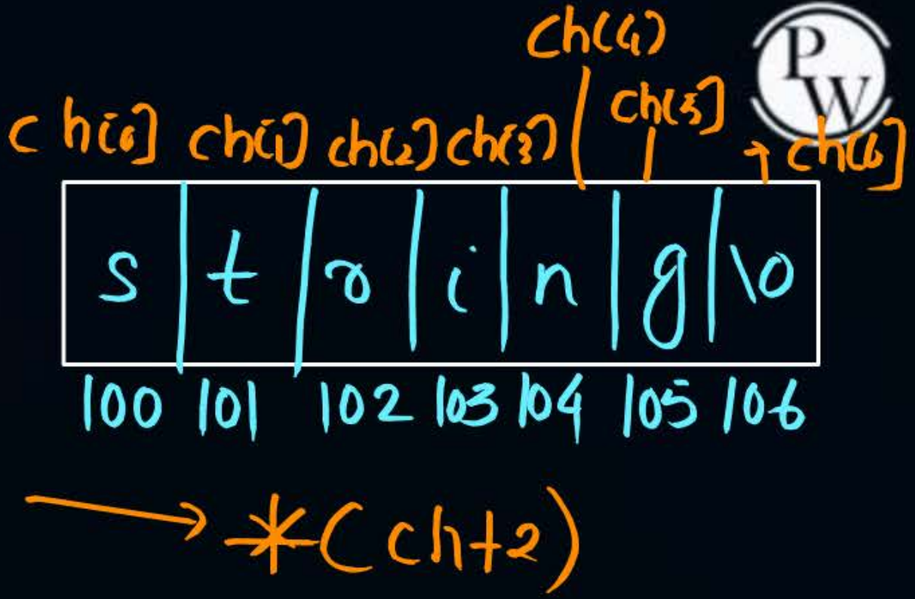
\0 - NULL character

ASCII value 0

Slide

String

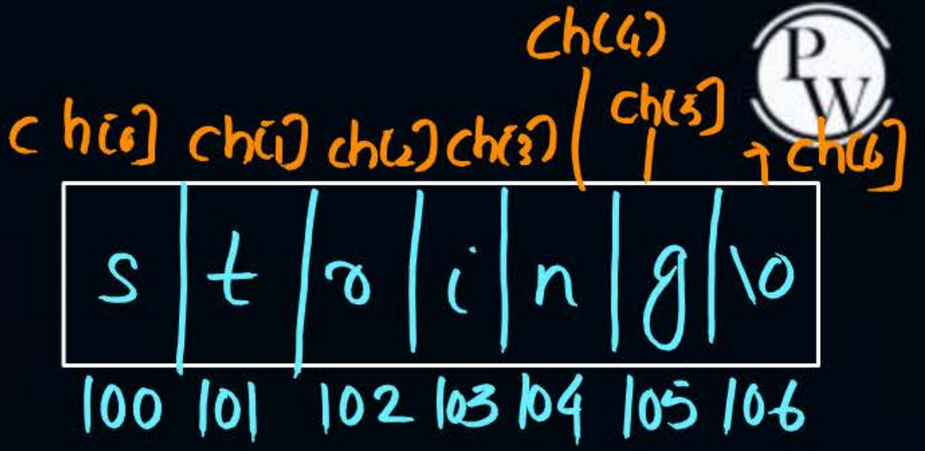
```
char ch[] = "String";
printf("%c", ch[2]);
```



→ *(ch+2)

String

```
char ch[] = "String";
```

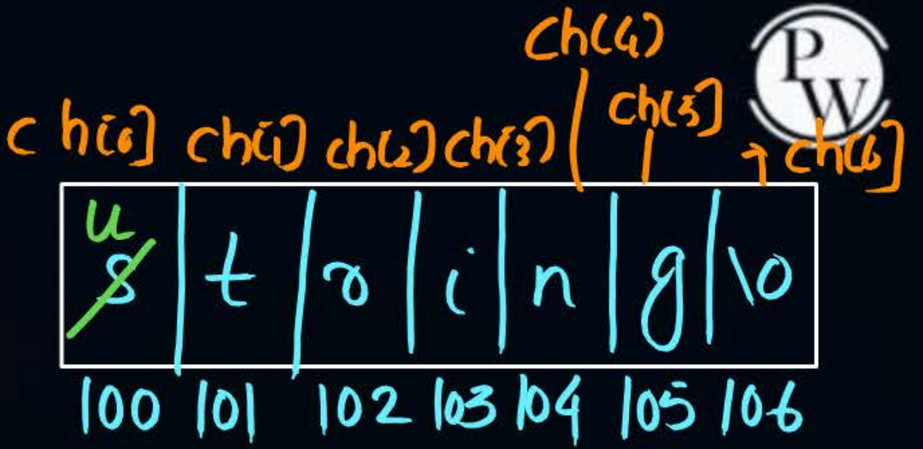



```
char *ptr;
ptr = ch;
printf("%c", *ptr++); // S
printf("%c", *ptr); // t
```

Slide

String

```
char ch[] = "String";
```



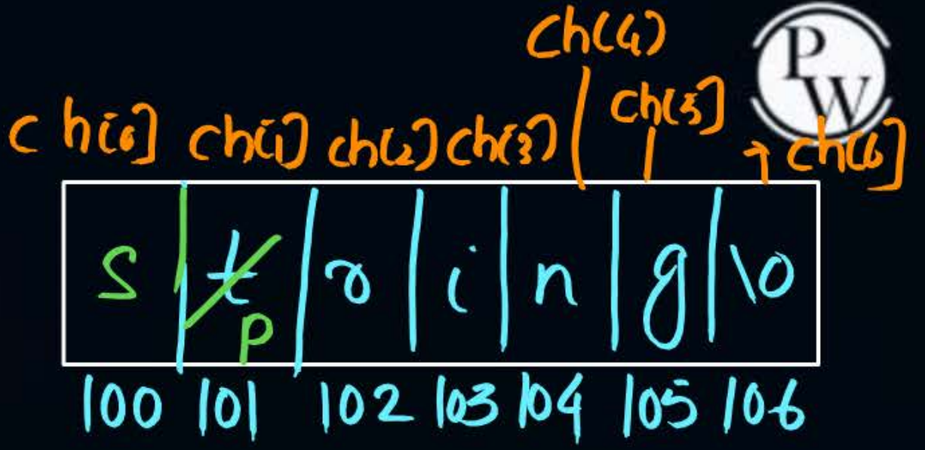

```
char *ptr;
ptr = ch;
(*ptr)+=2;   // (*ptr) = *ptr + 2
printf("%c", *ptr); // u
```

ptr | 100

Slide

String

```
char ch[] = "String";
```




```
ch[1] = 'p';
printf("%s", ch); //
```

ptr | 100

```
Spring
*(ch+1) = 'p';
printf("%s", ch+6); // Not print anything
```

Slide

String

printf % print everything till it gets the NULL character

```
char ch[] = "string";
ch[0] = '0';
printf("%s", ch);
```

0string

'0' - 0 character 48

0 NULL ASCII 0

Slide

String

printf % print everything till it gets the NULL character

```
char ch[] = "string";
```


```
ch[0] = 0;
printf("%s", ch);
```

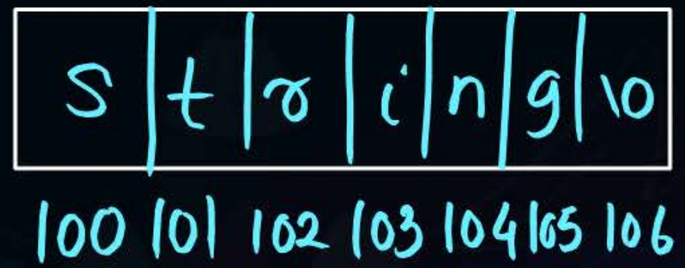
output :- No output

String

Second way to declare & initialize the string is using character pointer.

```
char *ptr = "string";
```

String stored in ROM



Slide

String

Character array string is mutable

Character pointer string is immutable

String

String maintained with character pointer allow to modify

```
char *ptr = "string";

ptr[1] = 'p';
printf("%s", ptr); // No output
```

---

base Address

```
ptr++; // allowed
printf("%s", ptr); // ring
```

Slide

String

```
char ch1[] = "string";
ch   ch2[] = "string";
```

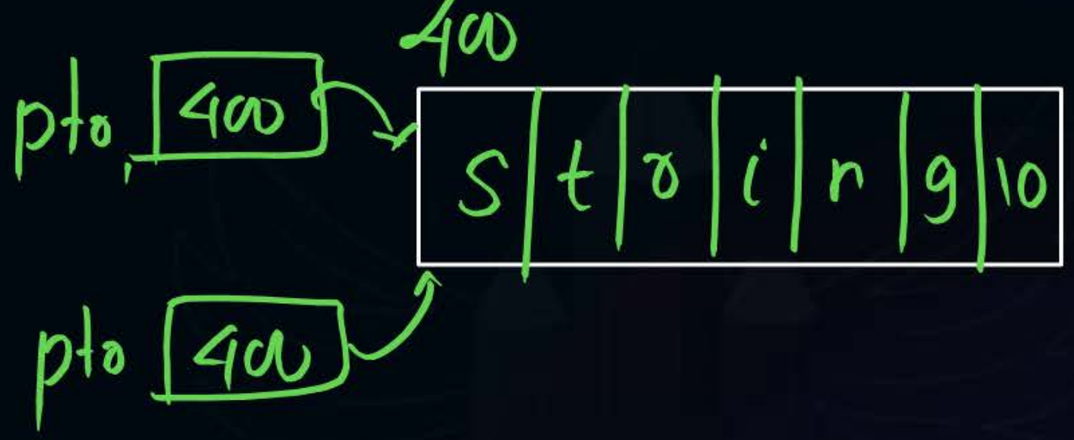
100

s	t	r	i	n	g	\0

200

s	t	r	i	n	g	\0

```
char *ptr1 = "string";
char *ptr2 = "string;
```



Slide

String

```
char ch[20];

ch = "string";

strcpy
```

```
char *ptr;

ptr = "string";
```

String

#include <string.h>

strlen( ) function take const character pointer

return unsigned value representing length of string

* Donot count the NULL character

```
char ch[]="string";
printf("%u", strlen(ch)); // 6
printf("%u", strlen(ch+3)); // 3
```

Slide

Question =

```
#include<stdio.h>
int main(int argc, char *argv[]){
char a = 'P';
char b = 'x';
char c = (a & b) + '*';
char d = (a | b) - '-';
char e = (a ^ b) + '+';
printf("%c %c %c\n", c, d, e);
return 0;
}
```

ASCII encoding for relevant characters is given below

*	+	–
42	43	45

(A) z K S
(B) 122 75 83
(C) * - +
(D) P x +

Slide

Question

#Q Consider the following C program segment:

```
char p [20];
char *s = "string";
int length = strlen(s);
for (i=0 ; i<length; i++)
        p[i] = s[length - i];
printf("%s",p);
```

The output of the program is

(a) gnirts

(b) string

(c) gnirt

(d) no output is printed

s[6-0] = s[6]

p[0]

p

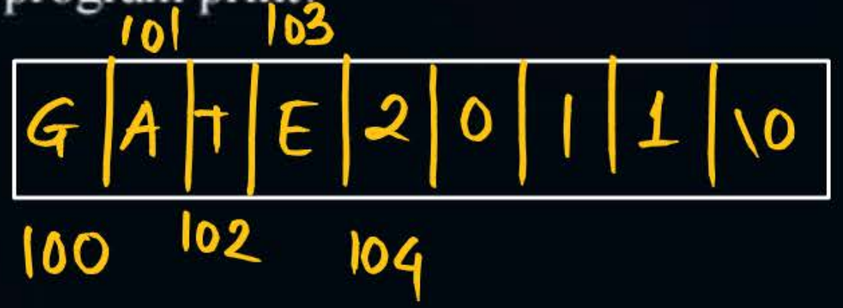
It prints every thing till Null

Slide

Question =

Q What does the following fragment of C-program print?

```
char c[]="GATE2011";
char *p =c;
printf("%s", p+p[3]-p[1]);
```



p 100

(A) GATE2011

(B) E2011

(C) 2011

(D) 011

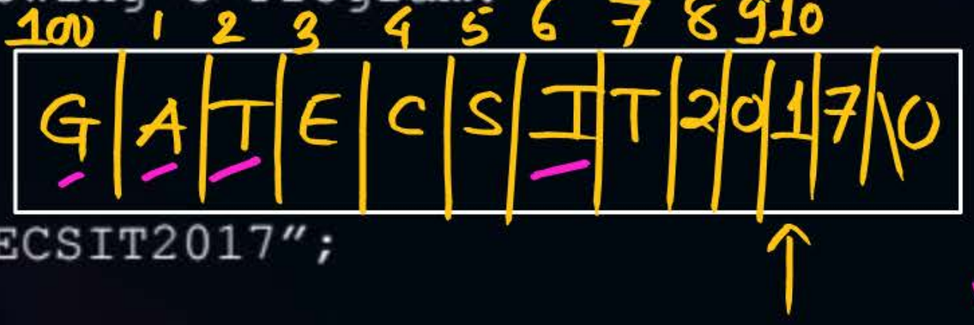
100 + 'E' - 'A'

100 + 69 - 65 = 104

Slide

Question

```
#Q Consider the following C Program.
#include <stdio.h>
#include< string.h>
int main () {
        char* c = "GATECSIT2017";
        char* p = c;
        printf("%d", (int) strlen (c+2[p]-6[p]-1));
        return 0;
}
```



The output of the program is _____2_____.

(A) 3

(B) 2

(C) 1

(D) 0

$100 + 2[p] - 6[p] - 1$

$100 + p[2] - p[6] - 1 = 100 + 'T' - 'I' - 1$

$= 100 + 84 - 73 - 1$

$= 110$

Slide

t.me/Abhishekshaarmapw

A - 65
B - 66
C - 67
D - 68
E - 69
F - 70
G - 71
H - 72
I - 73

a - 97
b - 98

Question

```
#Q Consider the following C Program.
#include <stdio.h>
#include< string.h>
int main () {
      char* c = "GATECSIT2017";
      char* p = c;
      printf("%d", (int) strlen (c+2[p]-6[p]-1));
      return 0;
}
The output of the program is _____.
```

a[i] = *(a+i) = *(i+a)

i[a]

a[c]

Slide

Question

HW

Consider the following C program:

```
#include<stdio.h>
void fun1(char *s1, char *s2){
        char *tmp;
        tmp = s1;
        s1 = s2;
        s2 = tmp;
}
void fun2(char **s1, char **s2){
        char *tmp;
        tmp = *s1;
        *s1 = *s2;
        *s2 = tmp;
}
```

```
int main (){
    char *str1 = "Hi",   *str2 = "Bye";
    fun1(str1,  str2);
    printf("%s %s ",  str1, str2);
    fun2(&str1,  &str2);
    printf("%s %s",  str1,  str2);
    return 0;
}
```

(A) Hi Bye Bye Hi

(B) Hi Bye Hi Bye

(C) Bye Hi Hi Bye

(D) Bye Hi Bye Hi

Slide

Question

Consider the following C program:

```
void abc(char*s)
{
        if(s[0]=='\0')return;
        abc(s+1);
        abc(s+1);
        printf("%c",s[0]);
}
main()
{       abc("123")
}
```

If abc(s) is called with a null-terminated string s of length n characters (not counting the null ('\0') character), how many characters will be printed by abc(s)?

(A) n

(B) 2n

(C) 2^n-1

(D) 2^n

Slide

2 mins Summary

Topic

Topic

Topic

Topic

Topic

String using char array

String using char pointer

Strlen().

Slide

THANK - YOU