

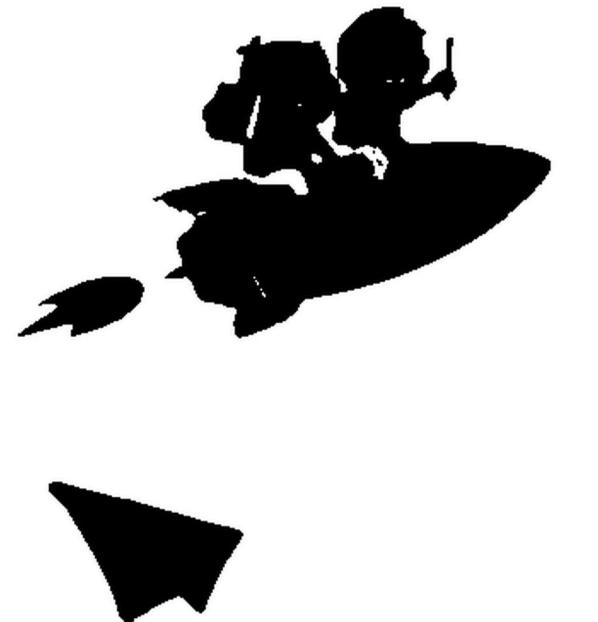
Computer Science & IT

C Programming

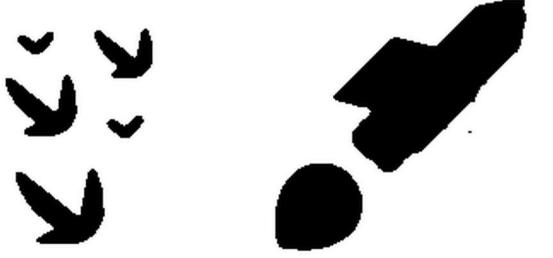
String

Lecture No. 01

By- Abhishek Sir



Recap of Previous Lecture



Topic

Topic

Topic

Topic

Topic

2D - 1-D array
problem

Topics to be Covered



Topic

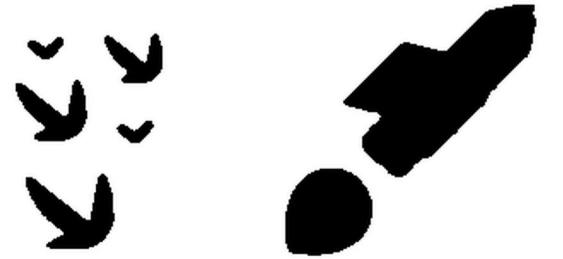
Topic

Topic

Topic

Topic

String
array of characters



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 characters

String is token "Hello World"

"
"
double quotes

String

char: LB



char ch[] = "string";

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

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

printf("%s", ch+2); // ring
Address first letter

0

\0 - NULL

character

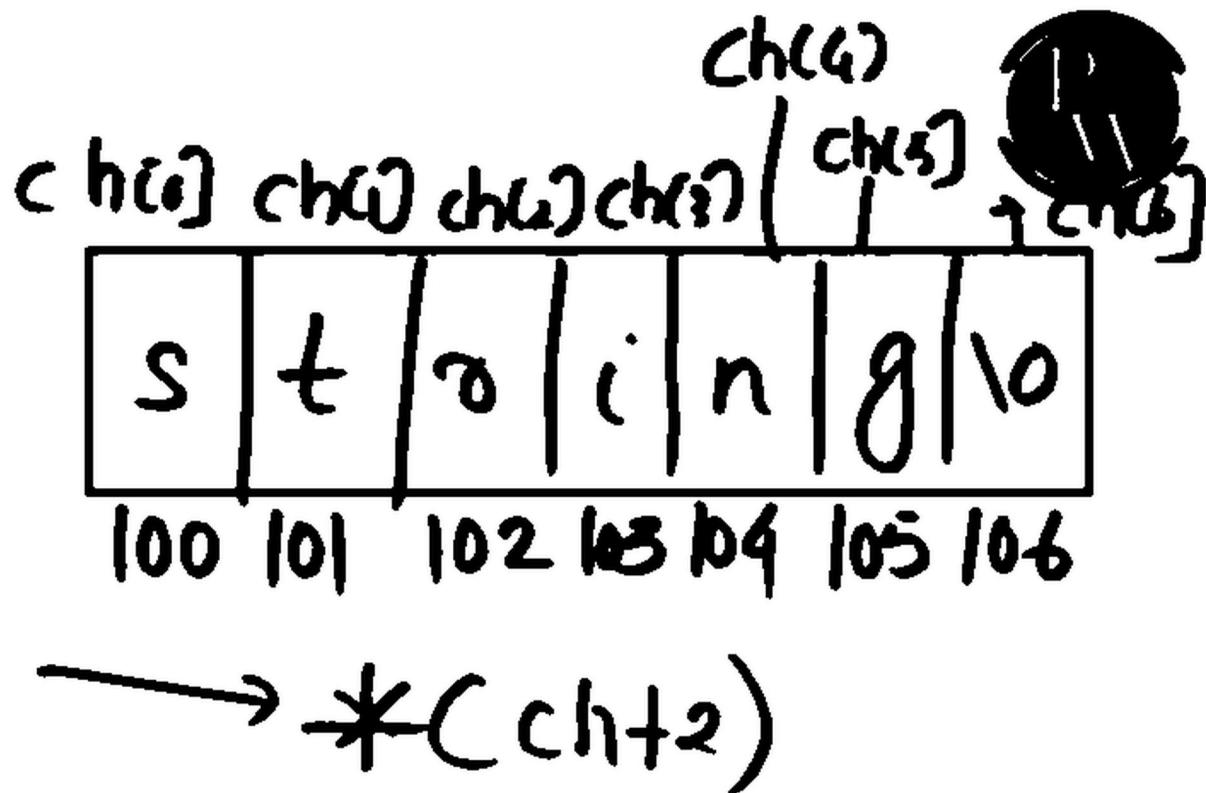
ASCII value

ch++; // Error Constant being Modified

String

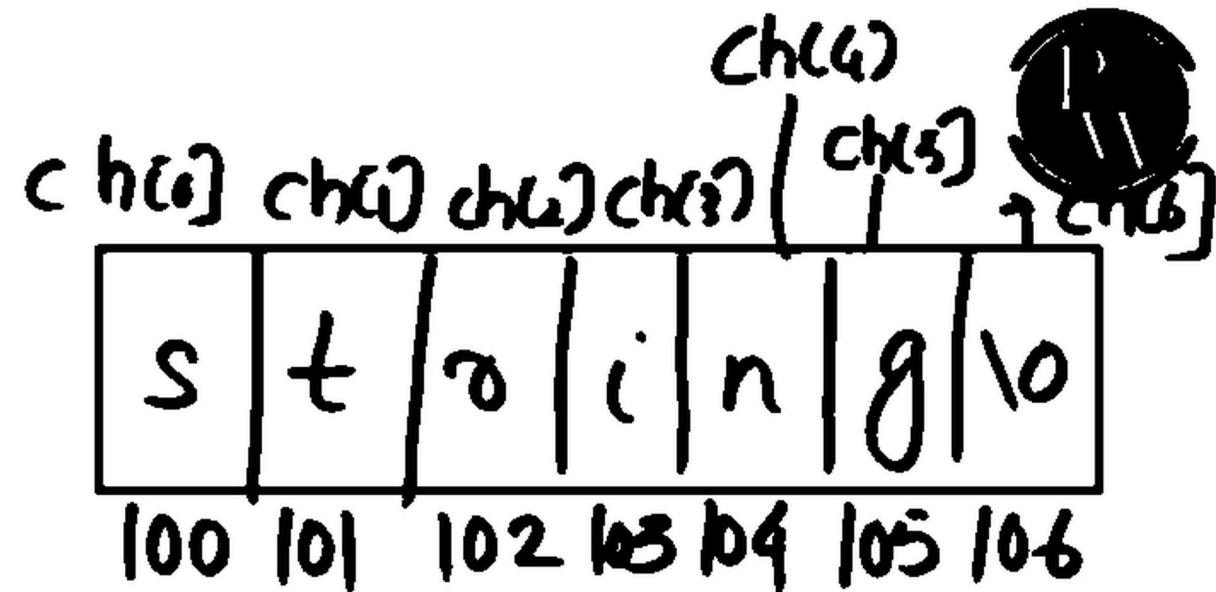
```
char ch[] = "String";
```

```
printf("%c.% ch[2]);
```



String

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

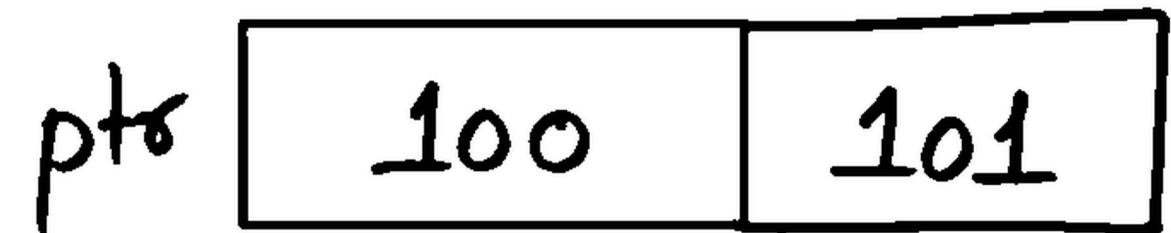


```
char * ptr;
```

```
ptr = ch;
```

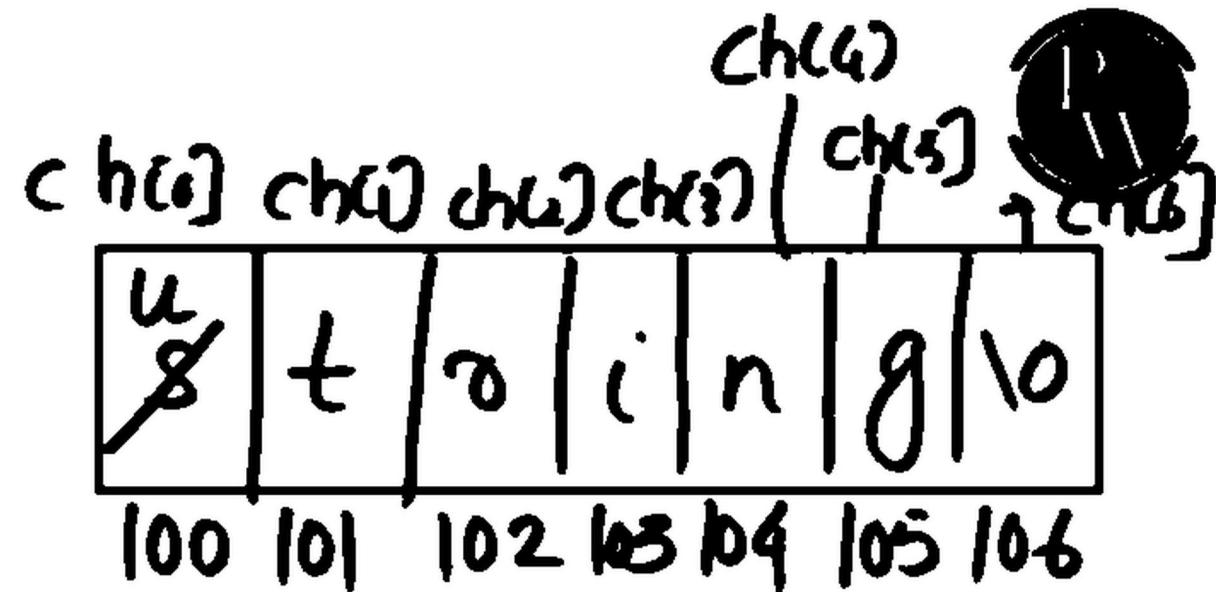
```
printf ("%c", *ptr++); // S
```

```
printf ("%c", *ptr); // t
```

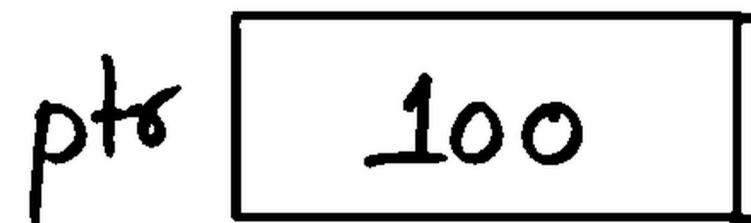


String

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



```
char * ptr;
```



```
ptr = ch;
```

```
(*ptr) += 2; // (*ptr) = *ptr + 2
```

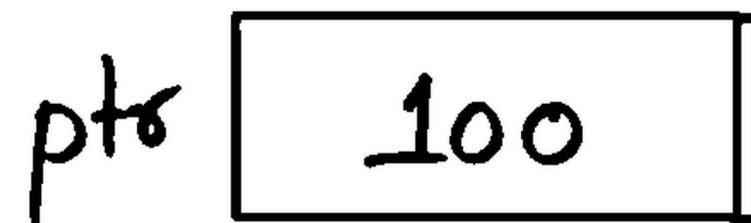
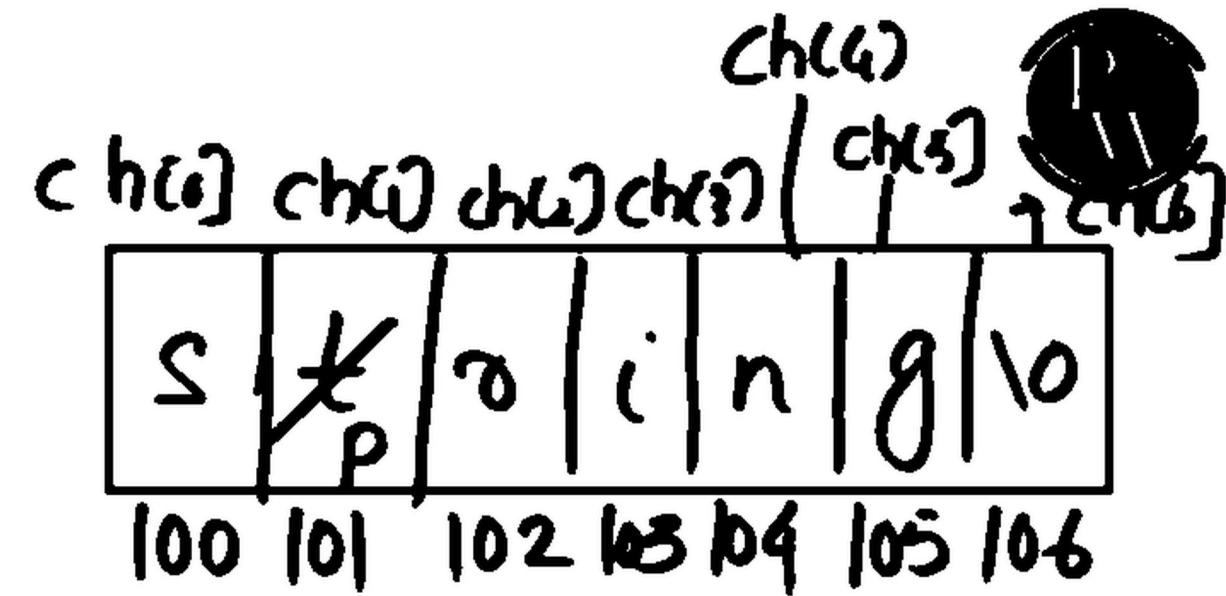
```
printf ("%c", *ptr); // u
```

String

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

```
ch[1] = 'P';
```

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



String $\ast(\text{ch} + 1) = 'P'$;

```
printf("%s", ch+6); // Not print anything
```

String

printf "%s" print everything till it gets the NULL character

char ch[] = "string";

O	s	t	r	i	n	g	\0
---	---	---	---	---	---	---	----

ch[0] = 'O';

printf ("%s", ch); Otring

'O' - 0 characters 48

0 NULL ASCII 0

String

printf "%s" print everything till it gets the NULL character

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

0						
\$	t	o	i	n	g	\0

```
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

S	t	r	i	n	g	\0
---	---	---	---	---	---	----

100 101 102 103 104 105 106

ptr [100]

String

Character array string is mutable

Character pointer string is immutable

String



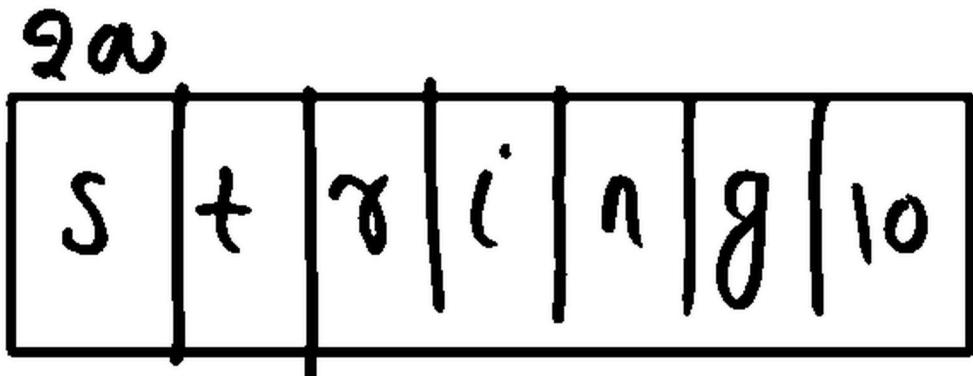
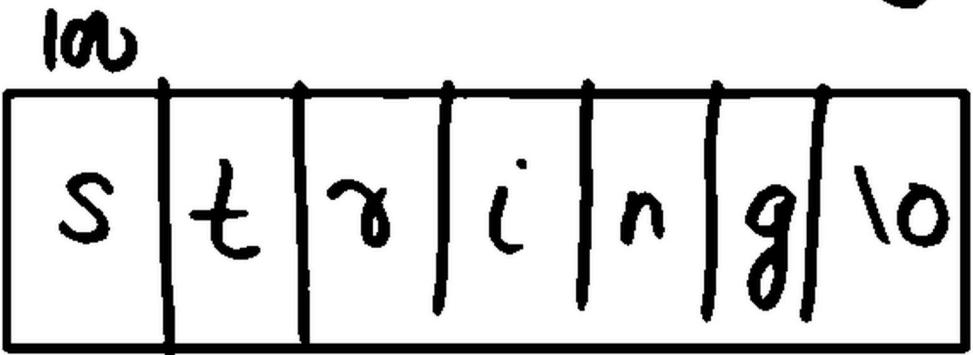
String maintained
with character pointer
allow to modify

```
char *ptr = "string";  
ptr[1] = 'P';  
printf("%s", ptr); // No output  
  
ptr++; // allowed  
printf("%s", ptr); // toring
```

String

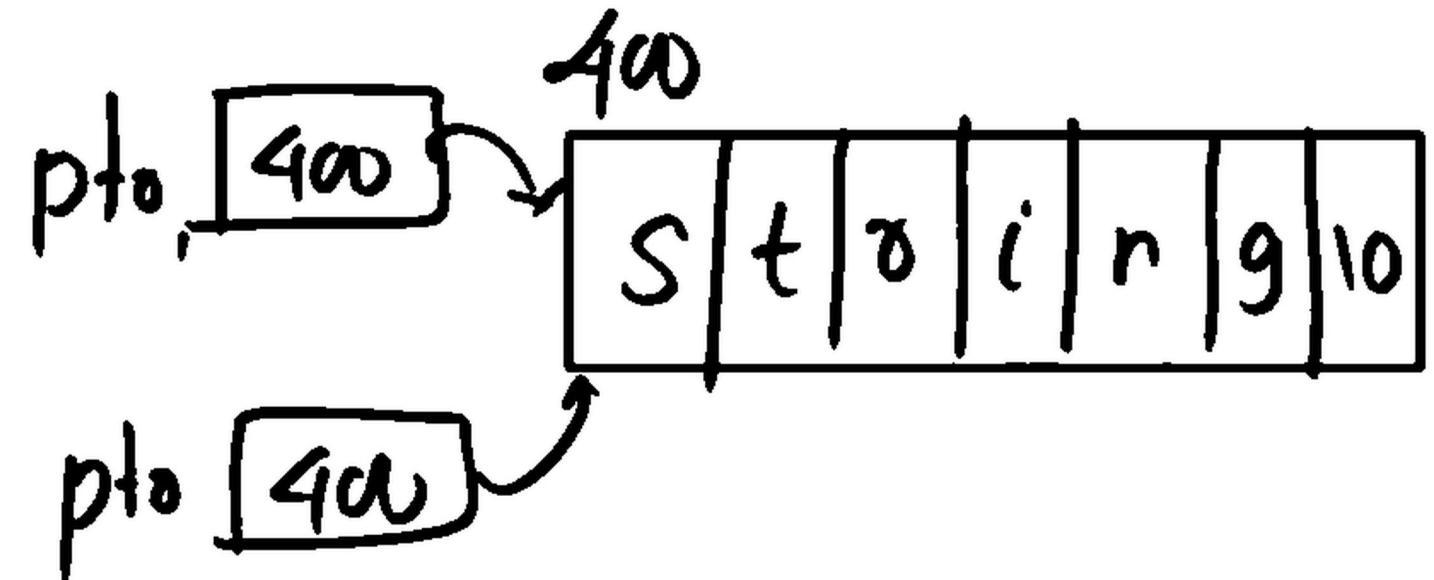
```
char ch[] = "string";
```

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



```
char *pto, = "string";
```

```
char *pto2 = "string";
```



String

char ch[20];

ch = "string";

strcpy



char *ptr;

ptr = "string";

String

```
#include  
<string.h>
```

stolen()

function take const character pointer

return

unsigned value representing length of string

* Donot count the NULL character

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

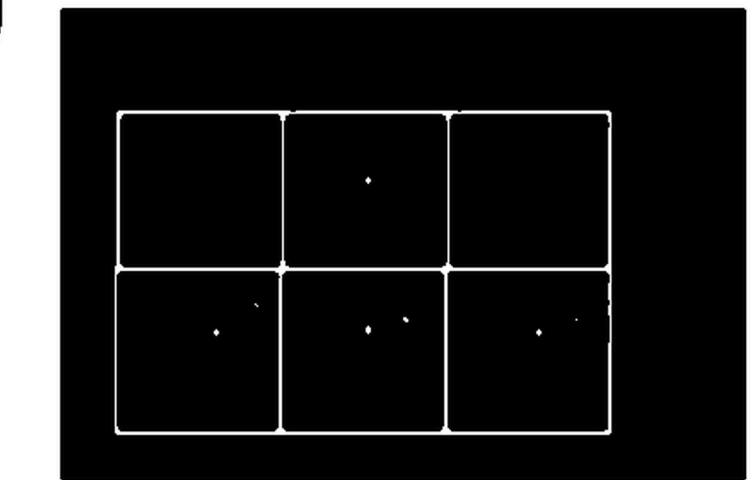
```
printf("%u", stolen(ch)); // 6
```

```
printf("%u", stolen(ch+3)); // 3
```

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;
}
```



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

ASCII encoding for relevant characters is given below
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);

```

It prints every thing till Null

The output of the program is

- (a) gnirts
- (b) string
- (c) gnirt
- (d) no output is printed

0	1	2	3	4	5	6
s	t	r	i	n	g	\0

$S[6-0] : S[6]$

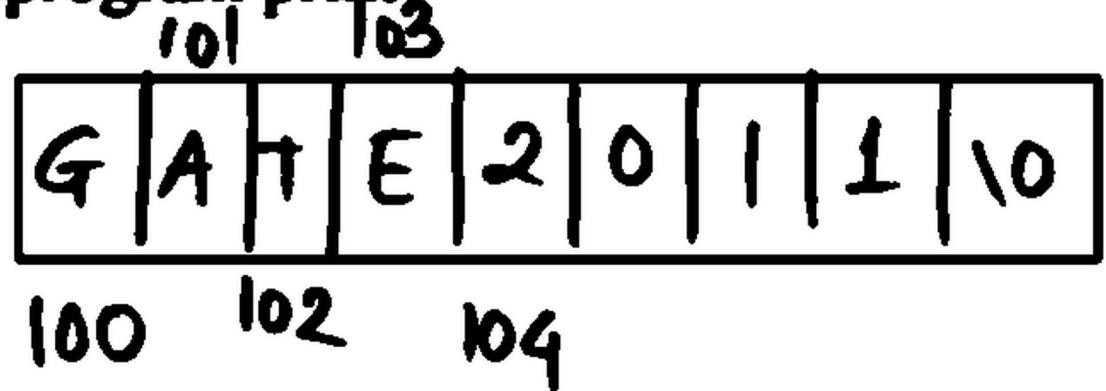
p(6)

\0						
----	--	--	--	--	--	--

Question =

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

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



(A) GATE2011

(B) E2011

~~(C) 2011~~

(D) 011

P 100

$$\begin{aligned} & 100 + \text{'E'} - \text{'A'} \\ & 100 + 69 - 65 = 104 \end{aligned}$$

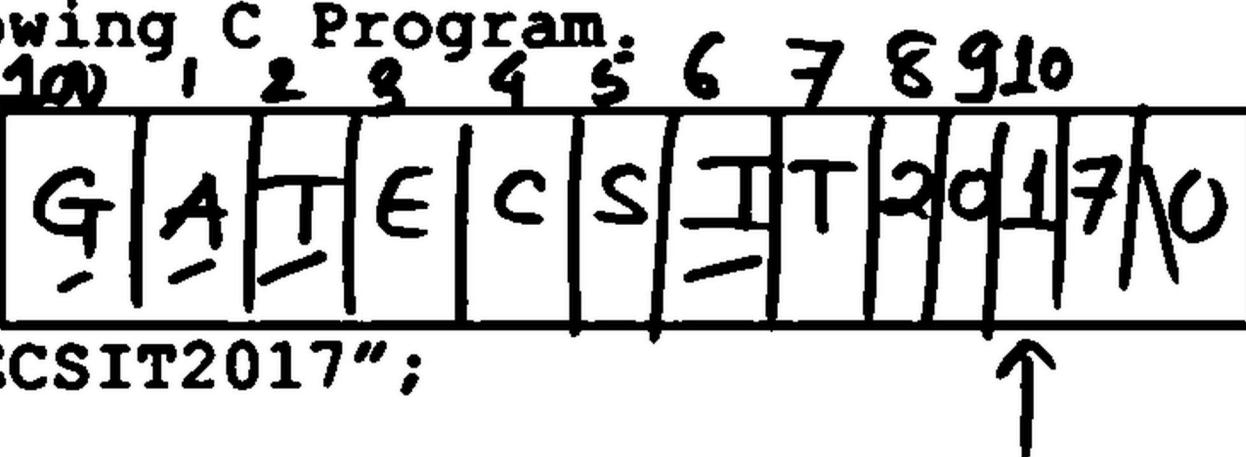
Question



#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) 3

(B) 2

(C) 1

(D) 0

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

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

$$= 100 + 84 - 73 - 1$$

$$= 110$$

t.me/Abhisheksharma pw

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 _____.

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

Question

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){ }
```

```
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

HW

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")
}
```

- (A) n
- (B) 2n
- (C) 2ⁿ⁻¹
- (D) 2ⁿ

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)?

2 mins Summary

Topic

String using char array

Topic

String using char pointer

Topic

Stolen(),

Topic

Topic

THANK - YOU