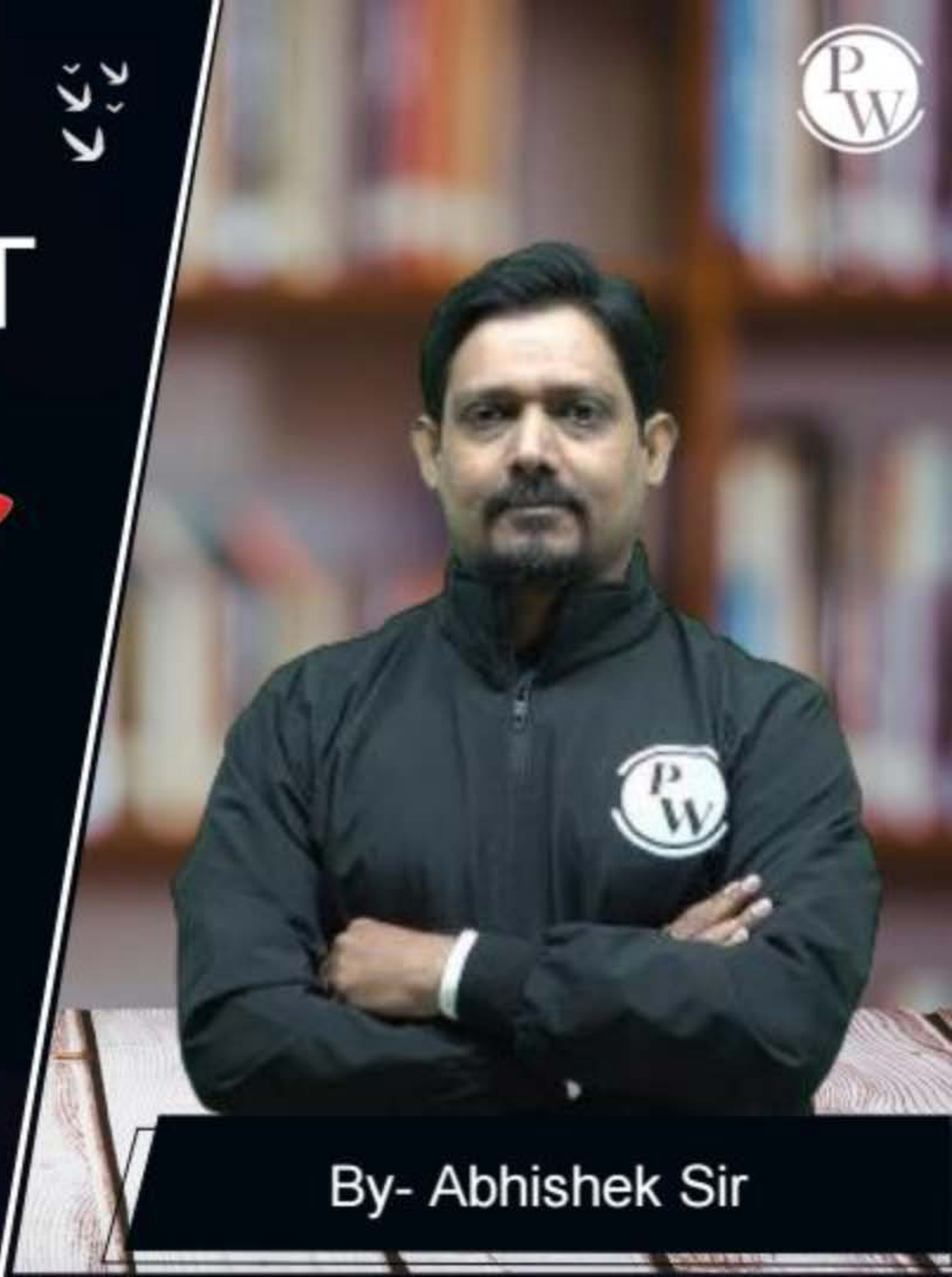
Computer Science & IT

C Programming

String in C programming

Lecture No. 02



By- Abhishek Sir

Recap of Previous Lecture

Topic — String

Topic — character array

Topic — character pointer

Topic — Array of string

Topic

Slide

Topics to be Covered

- Topic
- Topic
- Topic
- Topic
- Topic

practice problem

String function

Structure

Slide

Strlen

```
#include <stdio.h>
int main() {
    char *ch[2] = {"parakram", "vijay"};
    printf("%s\n",*ch);
    printf("%s\n",*ch+1);
    printf("%s\n",*ch+8);
    printf("%s\n",*(ch+1));
    printf("%c\n",ch[1][3]);
    return 0;
}
```

array

100 parakram

arakram

vijay

a

((ch+1)+3)

*(ch+1) = ch[1]

200	204
100	109
ch[0]	ch[1]

*(ch+1) ch[1]

p	a	r	a	k	r	a	m	\0	v	i	j	a	y	\0
100	101	102	103	104	105	106	107	108	109	110	111	112	113	114

Slide

Strlen

```
include <stdio.h>
int main(){
    char *ch[3] = {"parakram", "vijay","shreshth"};

    printf("%u\n", ch[0]);
    printf("%u\n", ch[1]);
    printf("%u\n", ch[2]);

    return 0;
}
```

100
109
115

p	a	r	a	k	r	a	m	\0	v	i	j	a	y	\0	s	h	r	e	s	h	t	h	\0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	2	2	2	2
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3

String Function

#include <String.h>

* strlen() — length → Single parameter
* strcmp() — string comparison → Two string
* strcpy() — string copy → (Deshnation, string)
* strcat() — string Concatenation, Two string.

Strlen

```
const char *ptr
```

```
char ch = 'a';
char *ptr;
ptr = &ch;
```

Constant char ←

```
const char *ptr1;
```

```
ptr1 = &ch;
*ptr = 'b', allowed
*ptr1 = 'b', ← Not allowe
```

Slide

Strlen

* strlen(const char *str)
* length of string return ✓
* it does not count NULL character. ✓
* Unsigned value is return ✓

String

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

int main(){
    char a[4] = "string";
    printf("%lu", strlen(a));
    return 0;
}
```

warning + 4 output

char a[10] = "string";

printf("%lu", strlen(a)); - 6

pnnlf ("%lu", stolen ("string"); - 6

Slide

String

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

#include <stdio.h>
int main(void){
  unsigned int plus_one = 1;
  int minus_one = -1;

  if(plus_one < minus_one)
    printf("1 < -1");
  else
    printf("boring");

  return 0;
}
```

GATE 2004

#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);
```

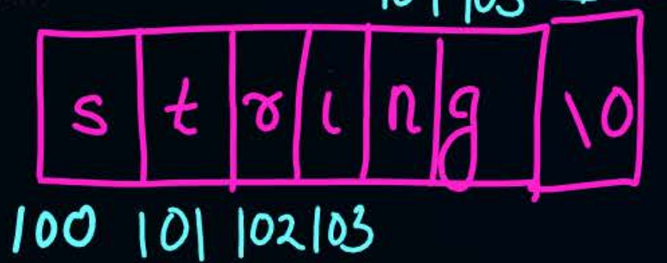
The output of the program is

(a) gnirts

(b) string

(c) gnirt

(d) no output is printed

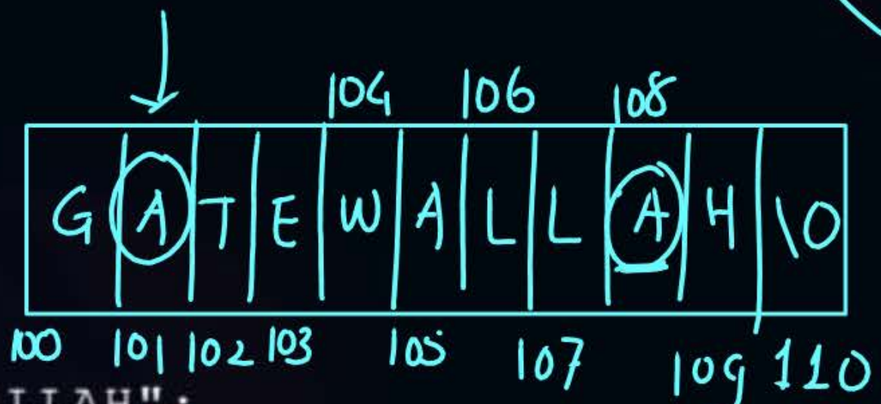


Slide

GATE 2011

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

```
#include <stdio.h>
#include<string.h>
int main(){
    char c[]="GATEWALLAH";
     char *p =c;
     printf ("%s", p+ p [8] - p [1]);
}
```



(A) Same string

(B) Different string

(C) TE WALLAH

(D) WALLAH

$p + p[8] - p[1]$

$100 + 65 - 65$

Slide

GATE 2004

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

#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 _____.

p[2] - p[6] - 1

84 - 73

(100 + T - I - 1)

2

105 106 105 (110)

(100 + 10)

G A T E C S I T 2 0 1 7 \0

100 101 102 103 104 107 109 111

(110)

Slide

A B C D E F G H I

65 70 73

J K L M N O P Q R S T

74 75 76 77 78 79 80 81 82 83 84

$\frac{x}{y}$

```
while(r>=y){
    r=r-y
    q=q+1
}
```

$\frac{7}{2}$

$r = x$

$y > 0$

$7-2=5$

1

$5-2=3$

2

$q=0$

Strcpy

```
strcpy( char *destination, const char *ptr)
```

↑ Destination

↑ Source

Const char *str

Char a[10];

a = "parakram";

Assign

Not allow

Strcpy

```
strcpy( char *destination, const char *ptr)
```

200

Char a[10],

Strcpy (a, "parakram");

Slide

Strcpy

Copy the character from source to destination

```
char a[100];

strcpy(a, “hello World”);

printf(“%s”, a);
```

String

```
#include <stdio.h>
#include<string.h>
int main(){
    char ch[20] ="parakram";
    strcpy(ch, "vijay");
    printf("%s", ch);
    return 0;
}
```

NULL character will copy

No of character printed

vijay\0

vijay

↓ 109

p	a	r	a	k	a	m	10	10	10
v	i	j	a	y	10	a	m	10	10	10

Slide

strcmp

strcmp (const char * s_1 , const char * s_2)

$s_1 > s_2$	positive value return	1
$s_1 == s_2$	0 will be printed	0
$s_1 < s_2$	Negative value return	-1

Slide

strcmp

```
#include<string.h>
#include<stdio.h>
int main(){

  char *s1 = "abc";

  char *s2 = "abc";

  printf("%d", strcmp(s1, s2)), ← this will print 0

  return 0;
```

Slide

strcmp

```
# include<string.h>
# include<stdio.h>
int main(){
    char *s1 = "abc"
    char *s2 = "abd"
    printf("%d", strcmp(s1, s2))
    return 0;
}
```

Each Index position ASCII value is compare

a b d

equal equal c < d

ASCII value

printf -1

$s_1 < s_2$

Slide

strcmp

```
# include<string.h>
# include<stdio.h>
int main(){

  char *s1 = "b",

  char *s2 = "abd"

  printf("%d", strcmp(s1, s2))

  return 0;
}
```

b _ _
a b d

b > a

$s_1 > s_2$

It will print

Slide

strcmp

```
# include<string.h>
# include<stdio.h>
int main(){
    char *s1 = "B";
    char *s2 = "abd";
    printf("%d", strcmp(s1, s2));
    return 0;
}
```

66 (ASCII value)

97

(-31)

$S_1 < S_2$

-31

(66 - 97) = -31 ✓

Slide

strcat

```
char *strcat(char *dest, const char *src);
```

String concatination

char pointer return

```
char ch[20] = {"parakram"}
printf("%s", strcat(ch, "vijay"))
```

strcat

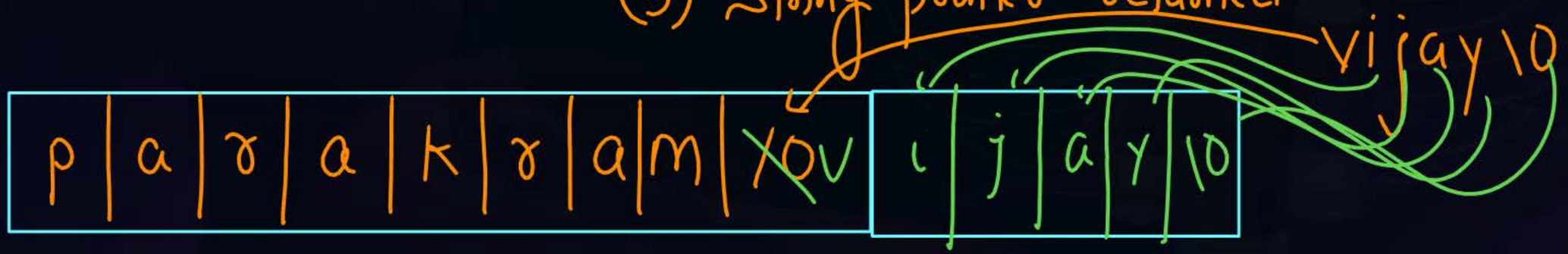
```
#include <stdio.h>
#include<string.h>
int main(){

    char ch[20] ="parakram";
    char *ch1 ="vijay";
    printf("%s", strcat(ch,ch1));
}
```

(1) ch will update

(2) NULL pointer for Destination overwritten by first character

(3) String pointer returned



Slide

2 mins Summary

Topic — String function

Topic — Strlen, strcpy

Topic — strcmp, strcat

Topic

Topic

Slide

THANK - YOU