Computer Science & IT

C Programming

String & structure

Lecture No. 01

By- Abhishek Sir

Recap of Previous Lecture

- Topic
- Topic
- Topic
- Topic
- Topic

practice problem · (1-D array)

Slide

Topics to be Covered

- Topic — String
- Topic — declaration of string
- Topic — Strlen() function
- Topic
- Topic

Slide

* string is array of character

* end marker . Last character '\0'

```
char ch[] = {'s','t','r','i','n','g','\0'};
```

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

String

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

        char ch[10] = "string";
        char *ch1 = "string";

        // printf the character
        printf("%c ", ch[4]);
        printf("%c\n", ch1[3]);

        // print the string format specifier %s, starting address
        // name of the array is address of first element
        printf("%s ", ch);
        printf("%s\n",ch1);

        printf("%s ", ch+2); // ring
        printf("%s",ch1+1); // tring
```

0 1 2 3 4 — 0 1 2 3

} two ways

% until it gets the NULL character

→ n

→ i

→ Address

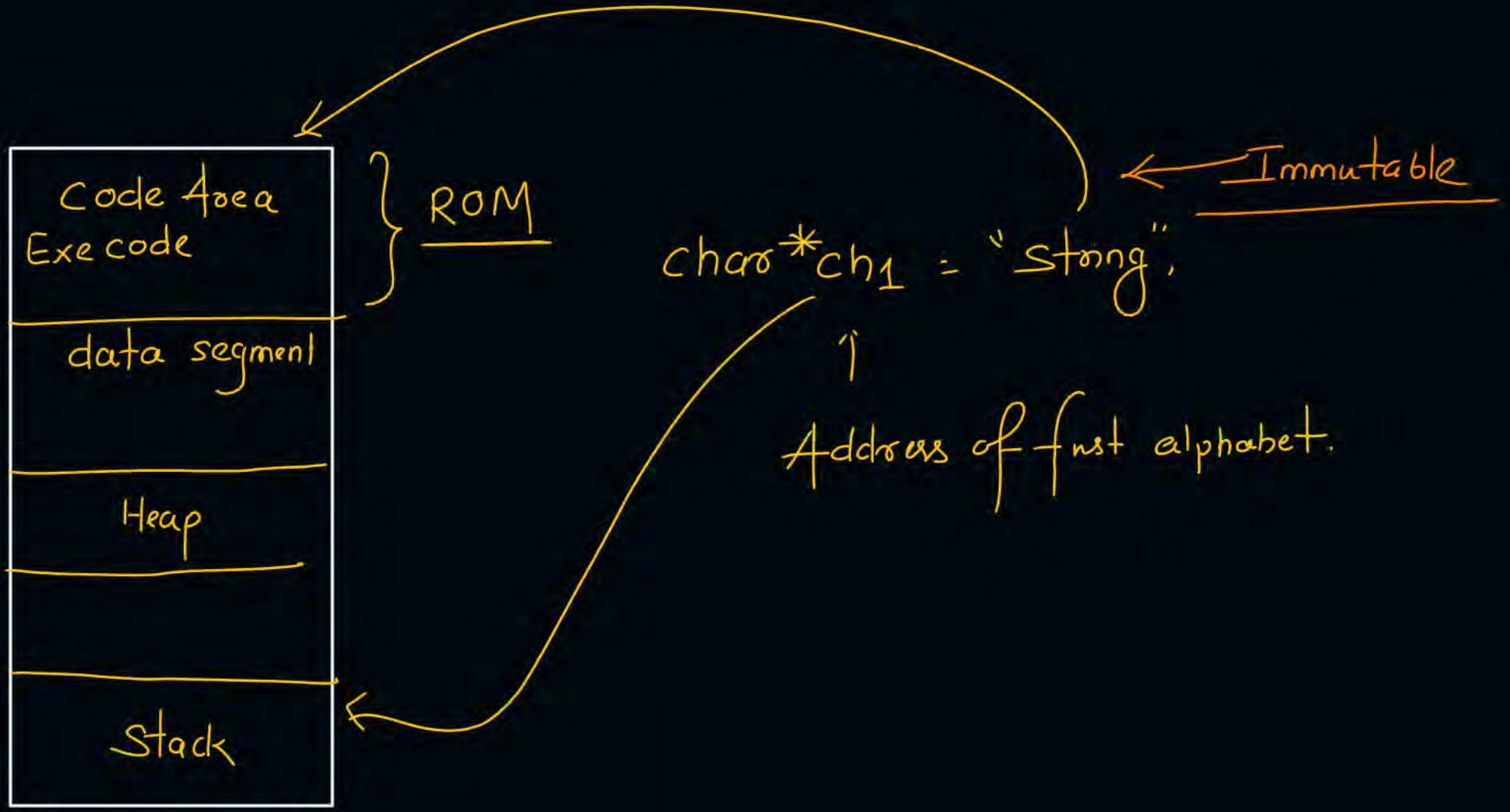
String

String

✓

✓

Slide


Code Area
Exe code
ROM
data segment
Heap
Stack
Immutable
char *ch1 = "strong";
Address of first alphabet.

String

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

        char ch[10] = "string";
        char *ch1 = "string";

        ch[1]='p';
        ch1[1]= 'p';
        printf("%s ", ch);
        printf("%s\n",ch1);

    }
```

"spring"

exit error code 139

Segmentation fault

Slide

String

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

        char ch[10] = "string";
        char *ch1 = "string";

        ch++;
        ch1++;
        printf("%s ", ch);
        printf("%s\n",ch1);

    }
```

Not allowed - Error (Array Base Address can't Increment)

allowed

String

```
#include <stdio.h>
    int main(void){
        char ch[10] = "string";
        char ch1[10] = "string";
        char *ch2 = "string";
        char *ch3 = "string";
        printf("%p ", ch);
        printf("%p\n",ch1);
        printf("%p ", ch2);
        printf("%p\n",ch3);
```

string

string

string\0

String

```
#include <stdio.h>
    int main(void){
        char ch[10];
        ch  = "string";
        char *ch2 = "string";
        printf("%s ", ch2);
        ch2 = "structure";
        printf("%s ", ch2);


    }
```

① ② ③ ④ ⑤ ⑥

ch[] = "string", wrong

② ⟶ Not allowed

Assignment of Multiple

(I) one after another

ch[0] = 's';

ch[1] = 't';

(II) strcpy

strcpy(ch , "string");

Slide

String

String Array Declaration

Memory Allocation:

When you declare a character array and initialize it with a string literal, the compiler allocates a contiguous block of memory on the stack (for local arrays) or in the data segment (for global/static arrays)

Pointer Declaration to a String Literal

Memory Allocation:

When you declare a character pointer and initialize it with a string literal, the string literal itself is stored in a read-only memory section (often the .rodata segment). The pointer variable, which stores the address of the first character of this literal, is allocated on the stack

String

String Array Declaration

Modifiability:

The characters within the array are modifiable. You can change individual characters or even the entire string content.

Pointer Declaration to a String Literal

Modifiability:

The string literal pointed to by the pointer is not modifiable. Attempting to modify it results in undefined behavior, often a segmentation fault.

String

String Array Declaration

Flexibility:

The size of the array is fixed at compile time and cannot be changed during runtime.

Pointer Declaration to a String Literal

Flexibility:

Pointers offer greater flexibility as they can be made to point to different memory locations (different strings) during runtime.

Question

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

```
#include<stdio.h>
void display(char *string){
printf("%s",string);
}
int main(){
char s[]="Hello World";
display(s+3);
return 0;
```

0 1 2 3

(A) Hello World

(B) llo World

(C) lo World

(D) o World

Slide

strlen(Address)

strlen("string") * length of string output

* unsigned value

* \0 character Not counted

Question

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

```
#include<stdio.h>
#include<string.h>
void display(char *s){
    for(int i =0;i<strlen(s);i++)
      s[i] = s[i]+1;}
int main(){
    char s[]="Hello";
    display(s+1);
  printf("%s", s);
  return 0;
}

};
```

s [1]

Hello
0 1 2 3 4

i=0;
i=1
i=2
i=3

H e l l o

H f m m p

(A) Hfmmp

(B) Hello

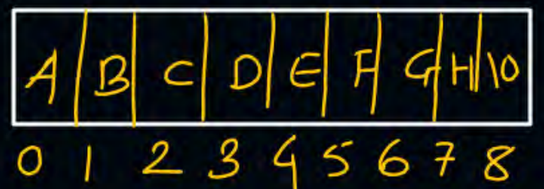
(C) Hlmmp

(D) Hfmmq

Slide

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

```
#include <stdio.h>
    int main(void){
        int k;
        char c[] = "ABCDEFGH";
        char * cp;
        for(cp = c+6; cp >= c+1;)
                printf("%c", * cp--);
        return 0;
}
```



(A) GFEDCB

(B) FEDCBA

(C) bcdefghi

(D) bcdffghi

G F E D C B

CP = 6 G >= 1 G

CP = 5 5 >= 1 F

CP = 4 4 >= 1

CP = 3 3 >= 1

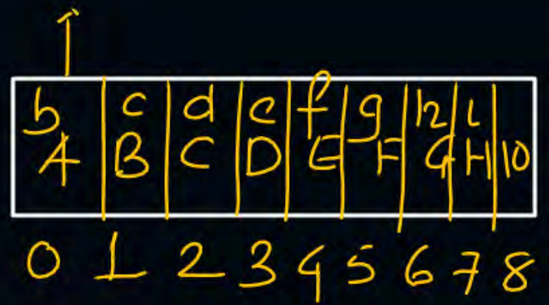
CP = 2 2 >= 1

1 >= 1

Slide

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

```
#include <stdio.h>
    int main(void){
        int k;
        char c[] = "ABCDEFGH";
        char * cp;
        for(k = 0; c[k] != '\0'; k++)
                    c[k] += 'b' - 'A';
        printf("%s\n", c);
        return 0;
}
```



(A) GFEDCB

(B) FEDCBA

(C) bcdefghi

(D) bcdffghi

c[0]: 65 + 98 - 65

66 + 98 - 65

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);
```

s

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

length = 6

s[6-5]
= s[1]

i=0 , i<6, i++

P[0] = s[6-0] = s[6]

\0	g	n	i	r	t					...

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]);
```

G	A	T	E	2	0	1	1	\0
0	1	2	3	4	5	6	7	8

(A) GATE2011

(B) E2011

(C) 2011

(D) 011

p → 0

p + p[3] - p[1] + 4

↓ output??

p + p[3] - p[1]

0 + 'E' − 'A'

0 + 69 − 65

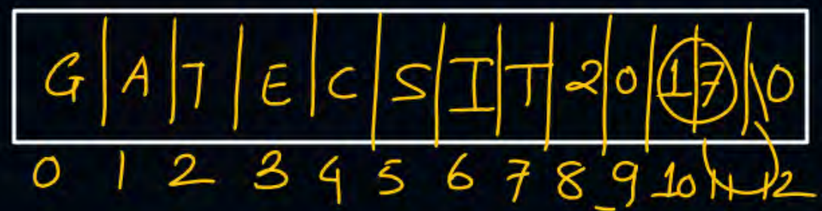
= 4

A	65
B	66
C	67
D	68
E	69

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 _____.
```



17 length

(0+'T'-'I'-1) = (0+84-73-1)

= 10

Slide

Question

#Q. Consider the following C program:

```
#include <stdio.h>
void stringcopy(char *, char *);
int main(){
    char a[30] = "@#Hello World!";
    stringcopy(a, a + 2);
    printf("%s\n", a);
    return 0;
}
void stringcopy(char *s, char *t)          {
    while(*t)
    *s++ = *t++;
}
```

Which ONE of the following will be the output of the program?

(A) @#Hello World!

(B) Hello World!

(C) ello World!

(D) Hello World!d!

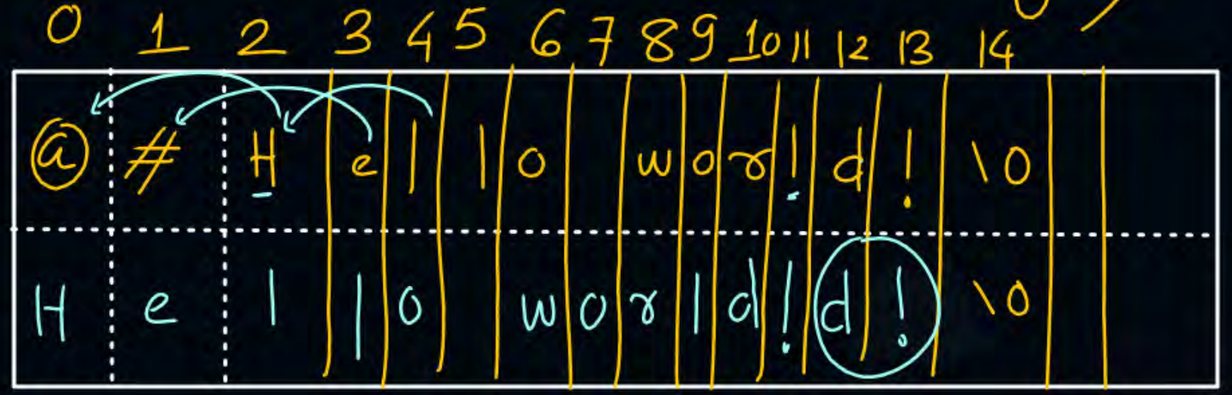
Slide

Question

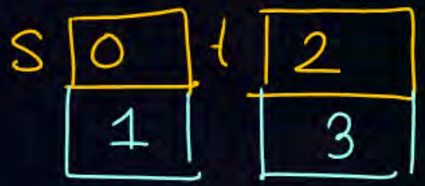
#Q. Consider the following C program:

```
#include <stdio.h>
void stringcopy(char *, char *);
int main(){
   char a[30] = "@#Hello World!";
   stringcopy(a, a + 2);
   printf("%s\n", a);
   return 0;
}
void stringcopy(char *s, char *t)
{
   while(*t)
   *s++ = *t++;
}
```

String copy (Source Memory, destination Memory)



a a+2



Which ONE of the following will be the output of the program?

Slide

Space ↔ Andhera Nahi

ASCII
32

`\0' - Andhera

`0'

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 _____.

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 _____.

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")
}
```

$T(0) = 0$

$T(n) = 2T(n-1) + 1$

(A) n

(B) 2n

(C) 2^n-1

(D) 2^n

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)?

Slide

Reduce to base condition

$n - k = 0 \Rightarrow n = k$

$2^n T(0) + 2^{n-1} + \cdots + 2 + 1$

$= 1 + 2 + 2^2 + \cdots + 2^{n-1}$

$\frac{1(2^n - 1)}{2 - 1} = 2^n - 1$

$T(0) = 0$

$T(n) = 2T(n-1) + 1$

$= 2(2T(n-2)+1) + 1$

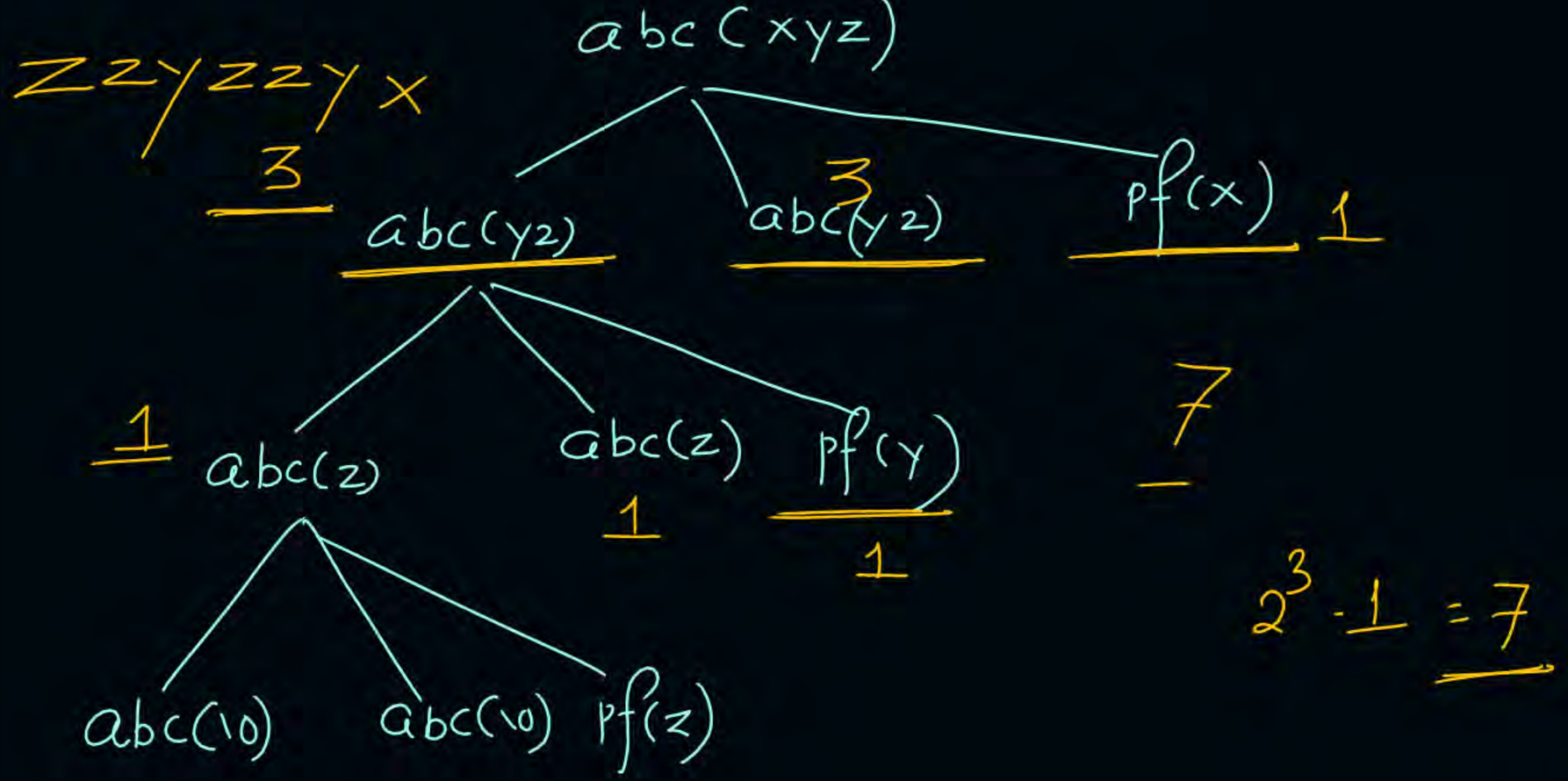
$= 2^2 T(n-2) + 2 + 1$

$= 2^2 (2T(n-3)+1) + 2 + 1$

$= 2^3 T(n-3) + 2^2 + 2 + 1$

k^{th} term

$= 2^k T(n-k) + 2^{k-1} + \cdots + 2 + 1$


zzyzzyx
3
abc(xyz)
abc(yz)
3
pf(x)
1
abc(yz)
1
abc(z)
abc(z)
1
pf(y)
1
7
abc(0)
abc(0)
pf(z)

$2^3 - 1 = 7$

2 mins Summary

Topic

char *strng immutable

Topic

char ch[] mutable

Topic

strlen() function, unsigned value

Topic

Topic

THANK - YOU

2-D
Array of
string
+
Structure