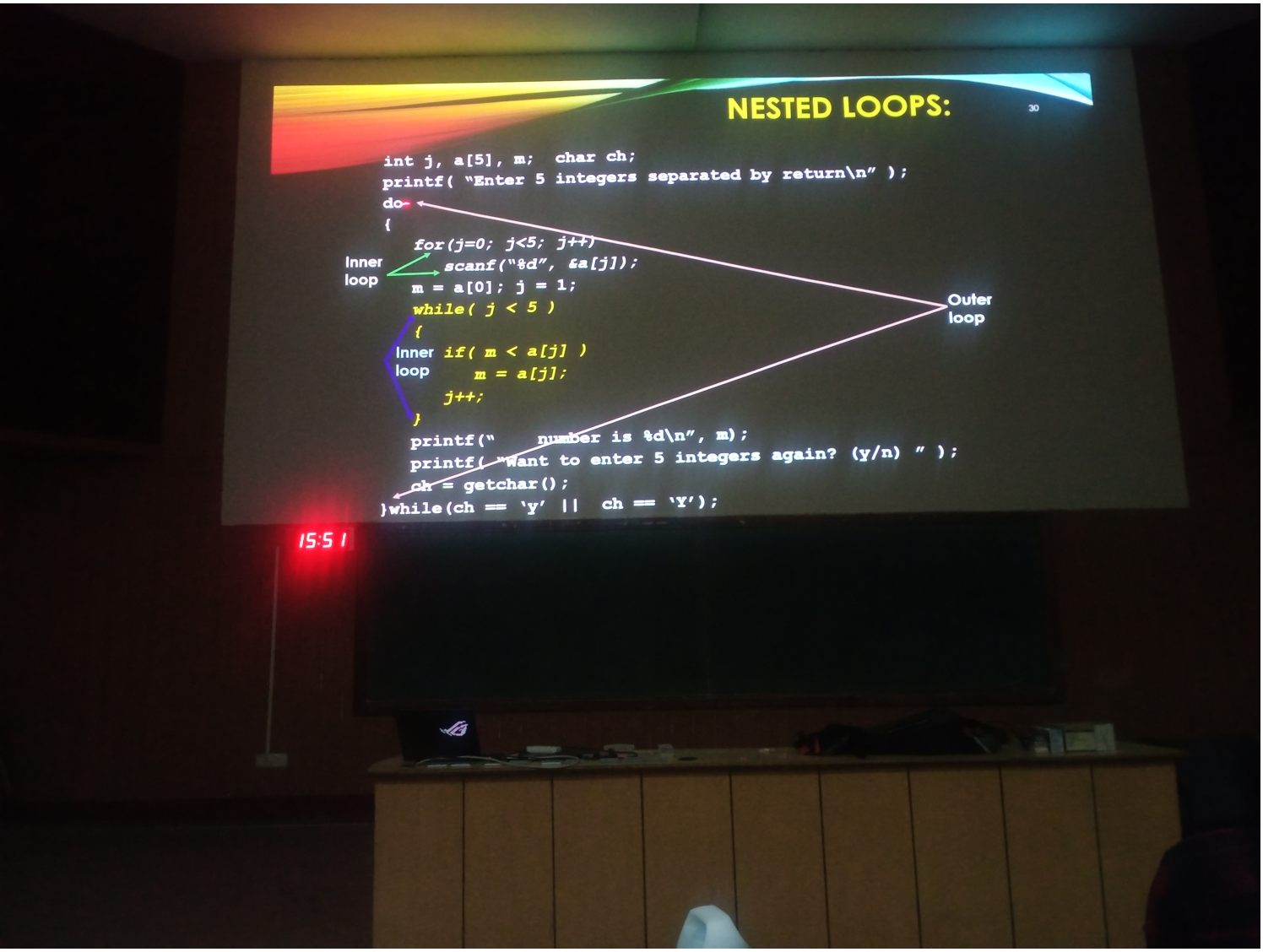
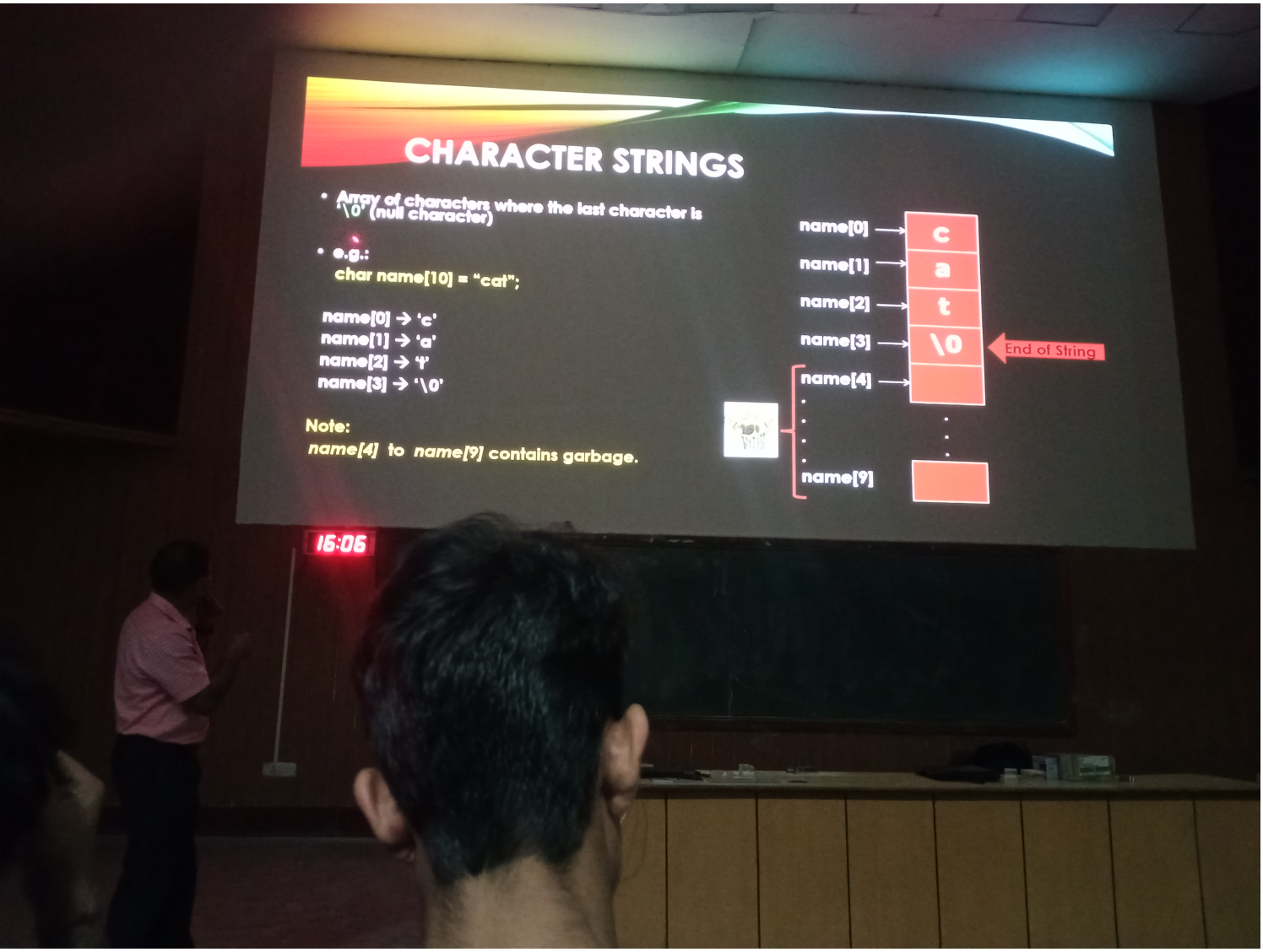
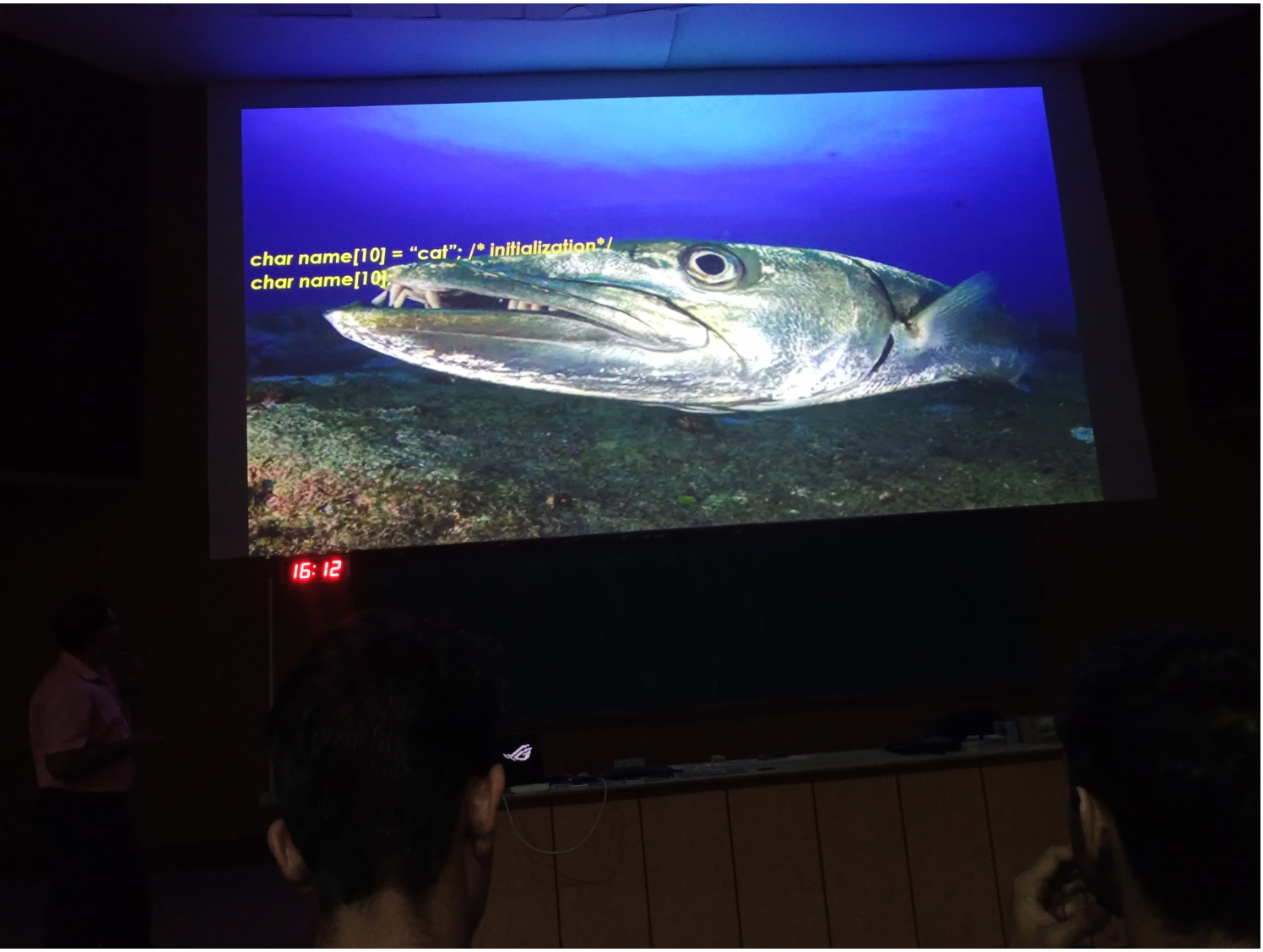
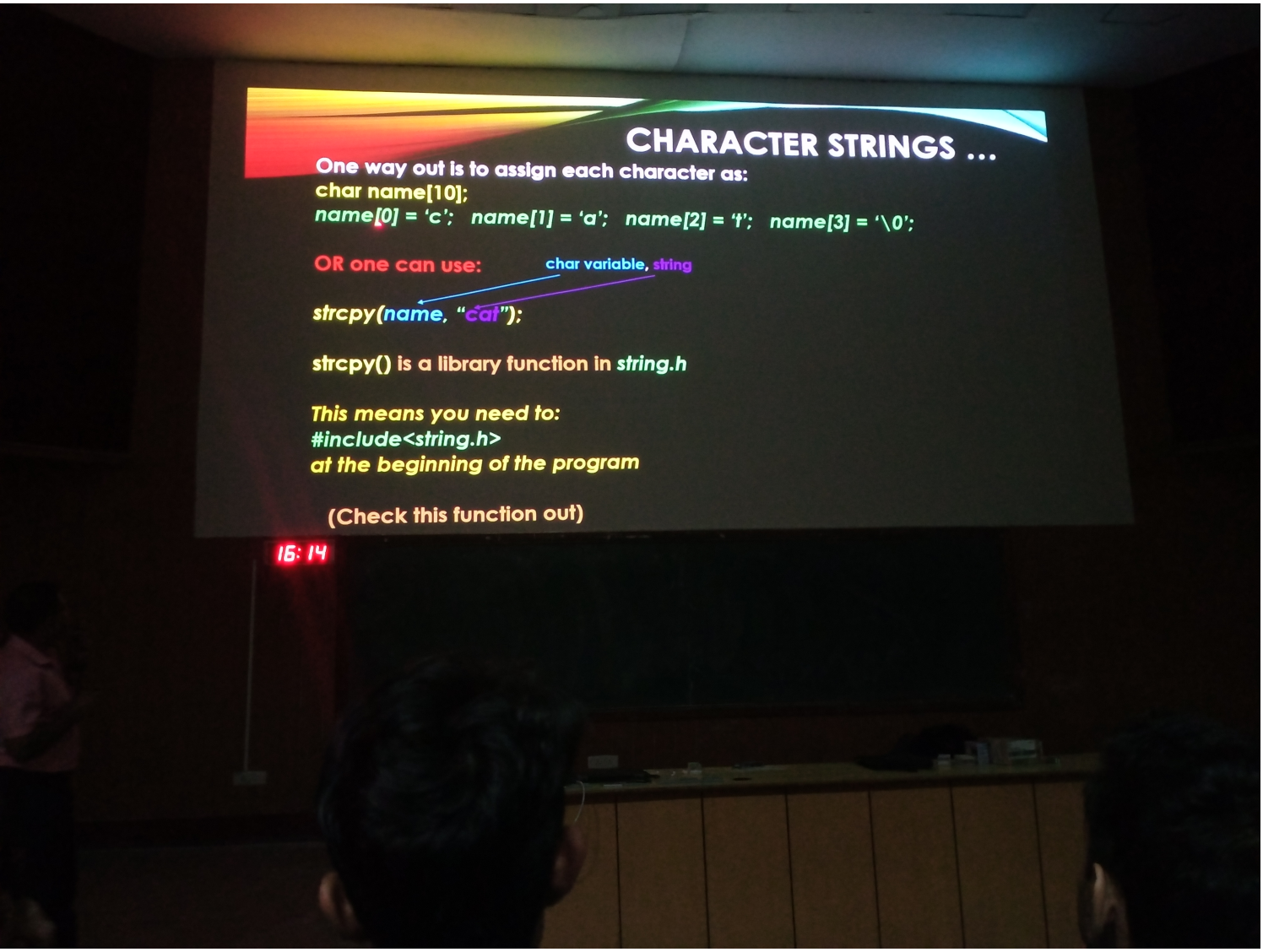
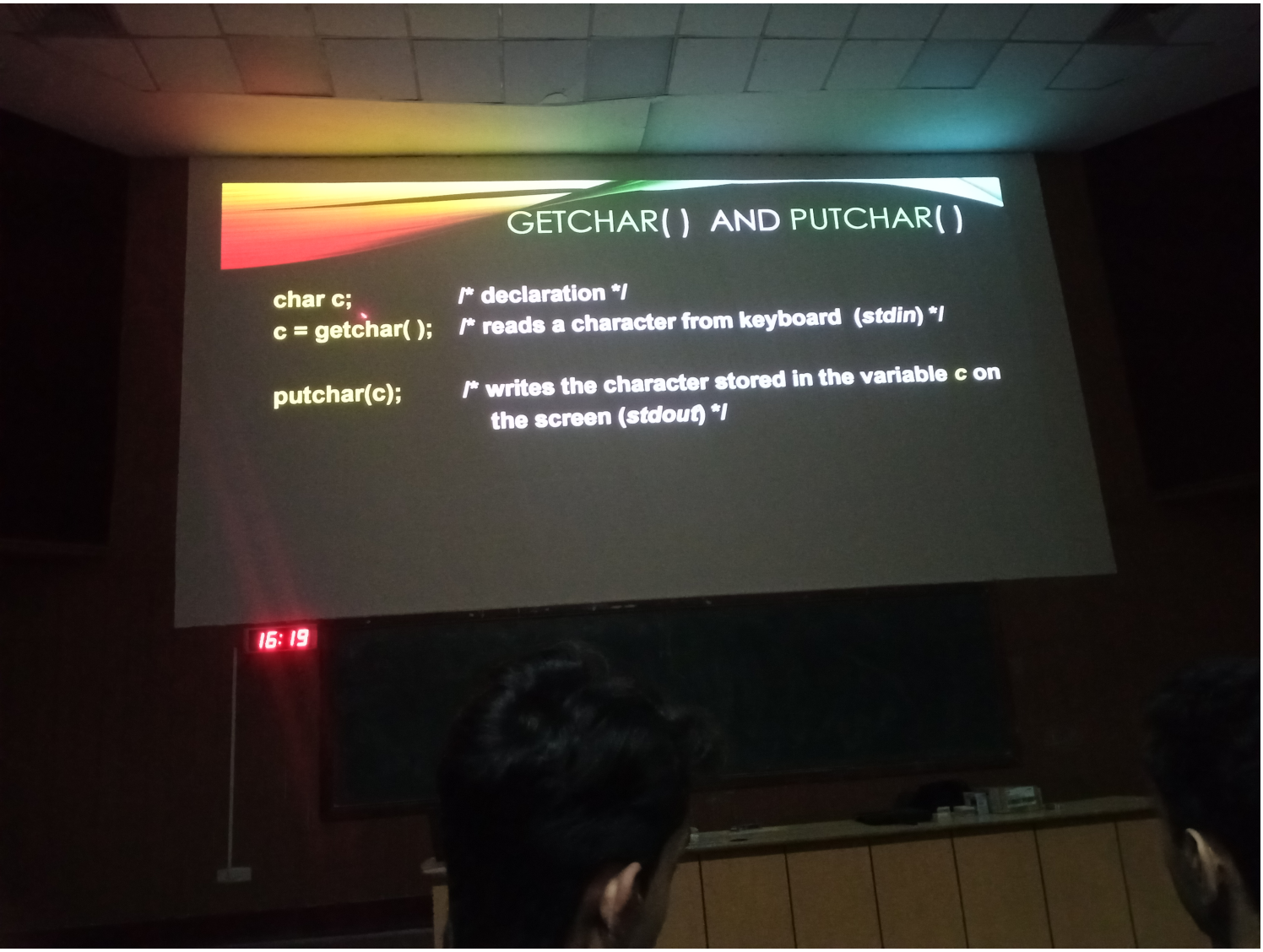
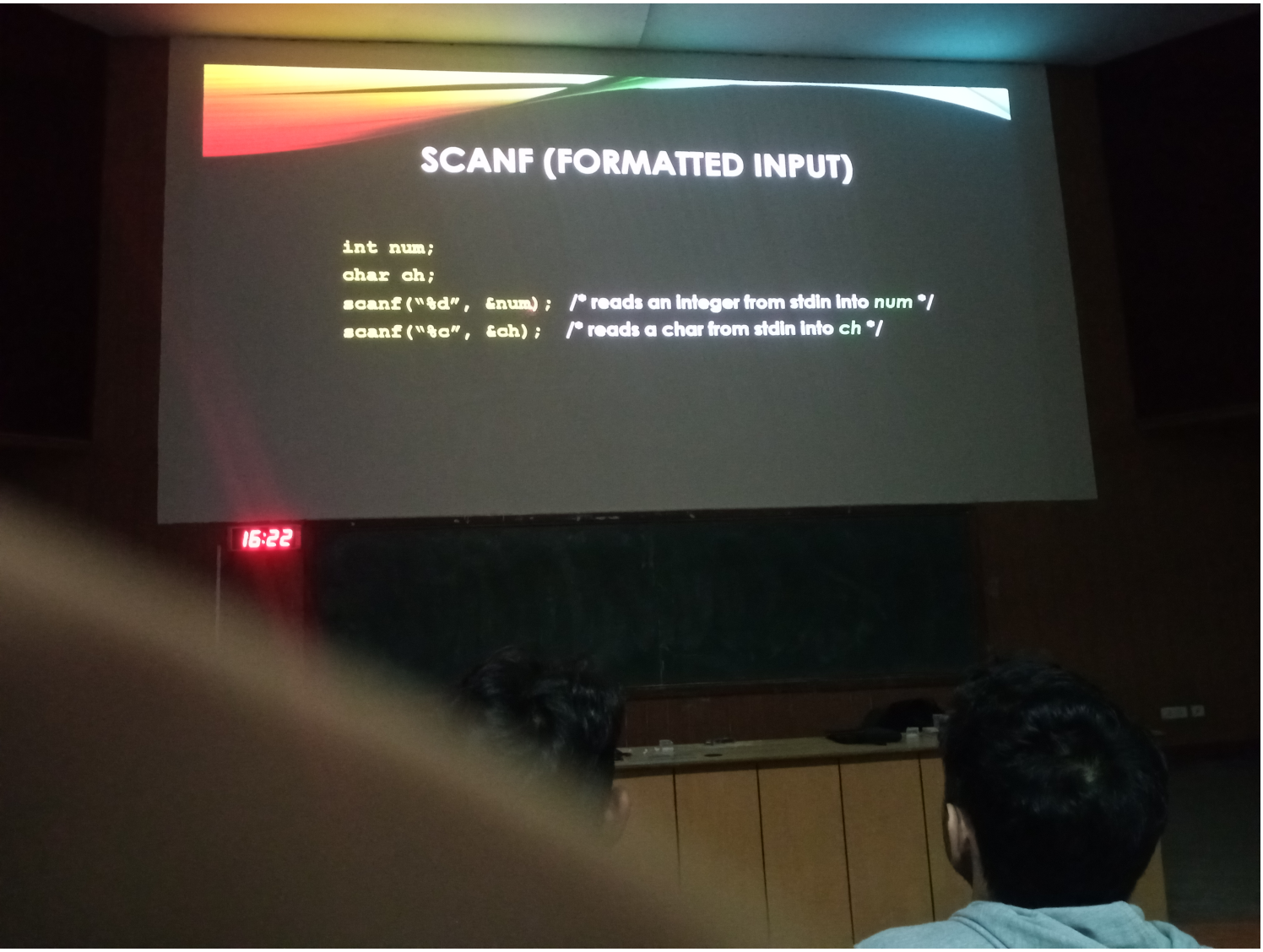

NESTED LOOPS:
30
int j, a[5], m; char ch;
printf("Enter 5 integers separated by return\n");
do
{
for(j=0; j<5; j++)
scanf("%d", &a[j]);
Inner loop
m = a[0]; j = 1;
while(j < 5)
{
Inner loop
if(m < a[j])
m = a[j];
j++;
}
Outer loop
printf(" number is %d\n", m);
printf("Want to enter 5 integers again? (y/n) ");
ch = getchar();
}while(ch == 'y' || ch == 'Y');

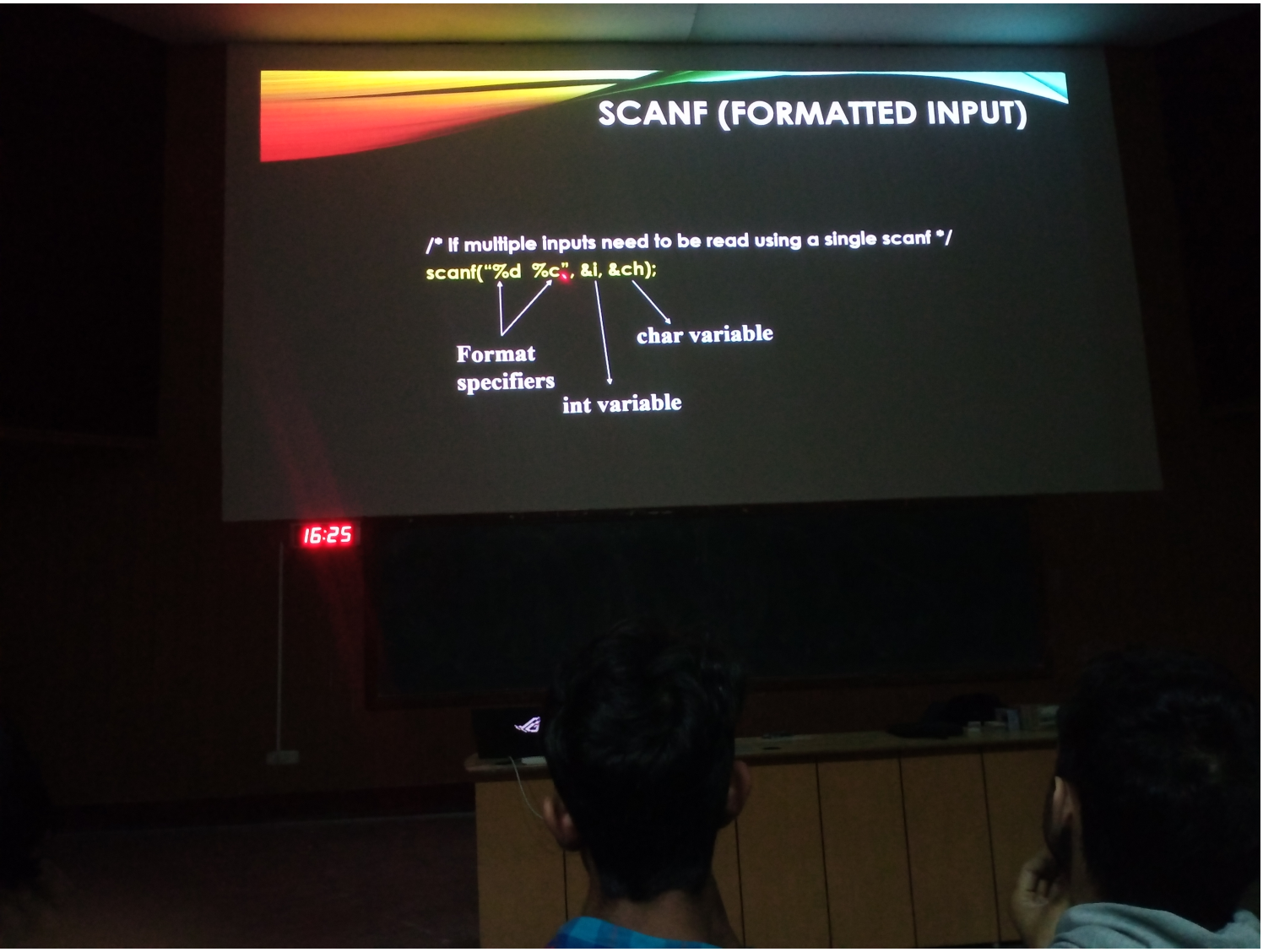

CHARACTER STRINGS
Array of characters where the last character is '\0' (null character)
e.g.:
char name[10] = "cat";
name[0] → 'c'
name[1] → 'a'
name[2] → 't'
name[3] → '\0'
Note:
name[4] to name[9] contains garbage.
name[0]
name[1]
name[2]
name[3]
name[4]
name[9]
c
a
t
\0
End of String
16:06


char name[10] = “cat”; /* initialization*/
char name[10];
16: 12

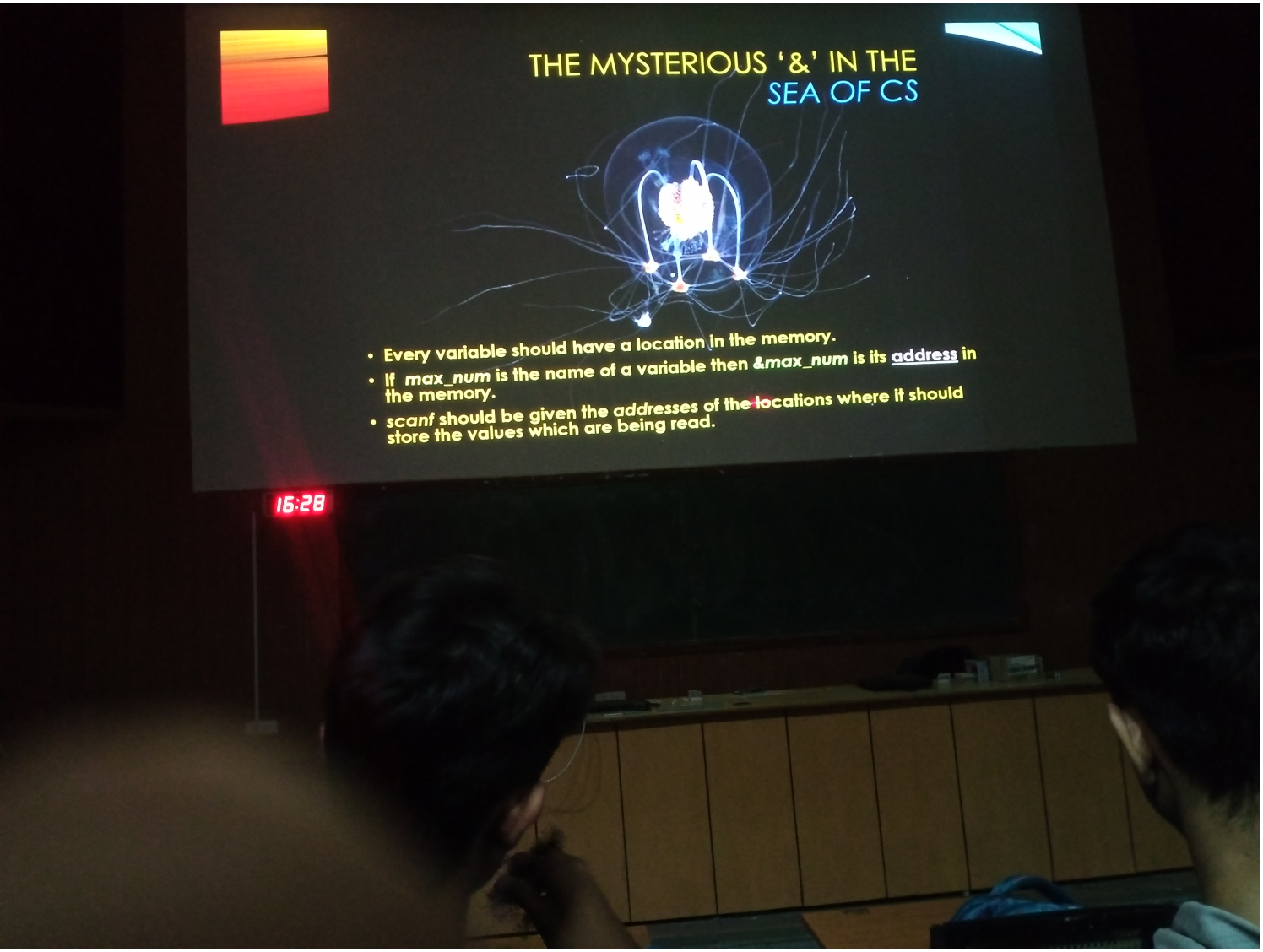

CHARACTER STRINGS ...
One way out is to assign each character as:
char name[10];
name[0] = 'c'; name[1] = 'a'; name[2] = 't'; name[3] = '\0';
OR one can use:
char variable, string
strcpy(name, "cat");
strcpy() is a library function in string.h
This means you need to:
#include<string.h>
at the beginning of the program
(Check this function out)
16:14

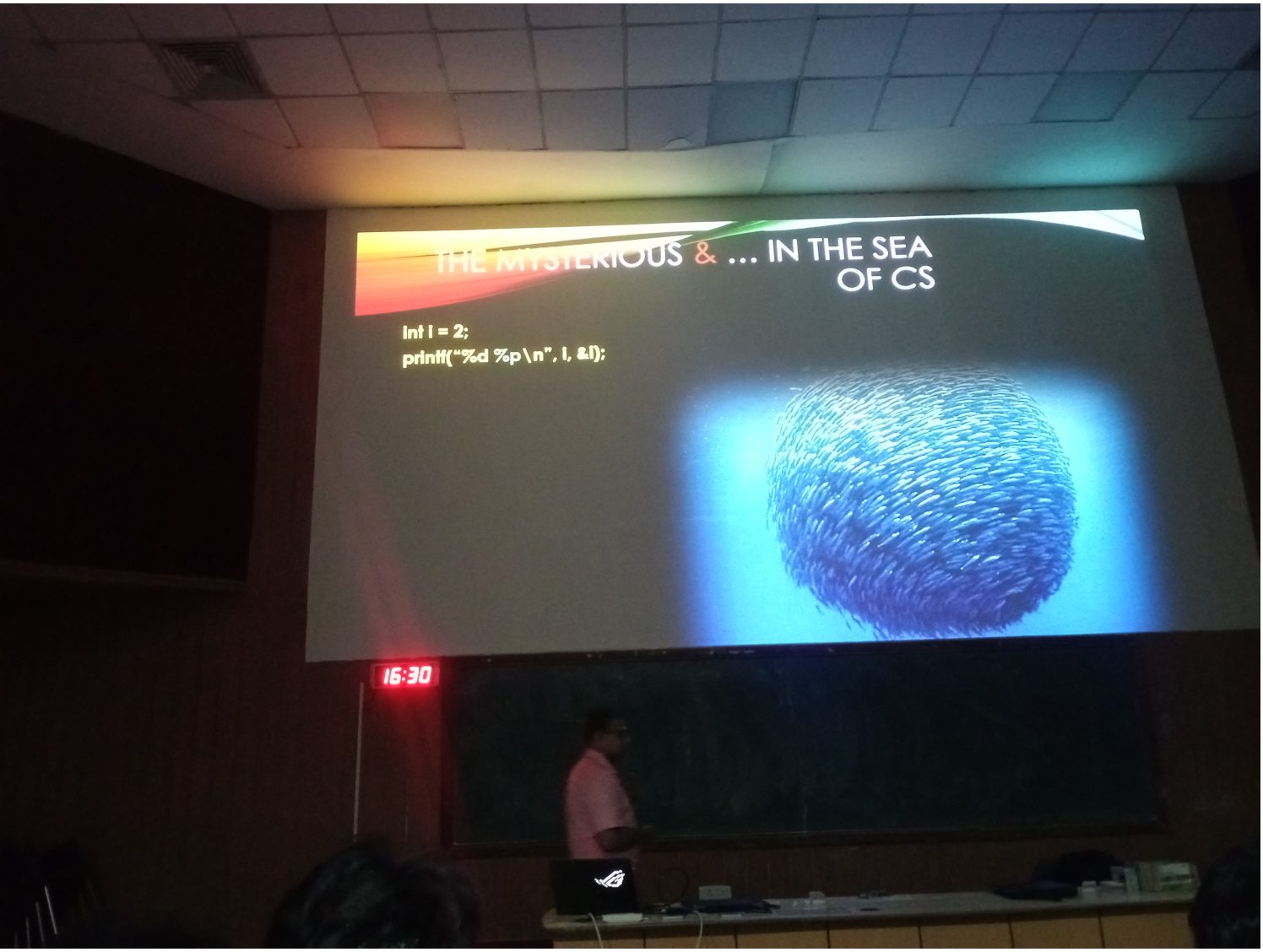

GETCHAR( ) AND PUTCHAR( )
char c; /* declaration */
c = getchar( ); /* reads a character from keyboard (stdin) */
putchar(c); /* writes the character stored in the variable c on the screen (stdout) */
16: 19

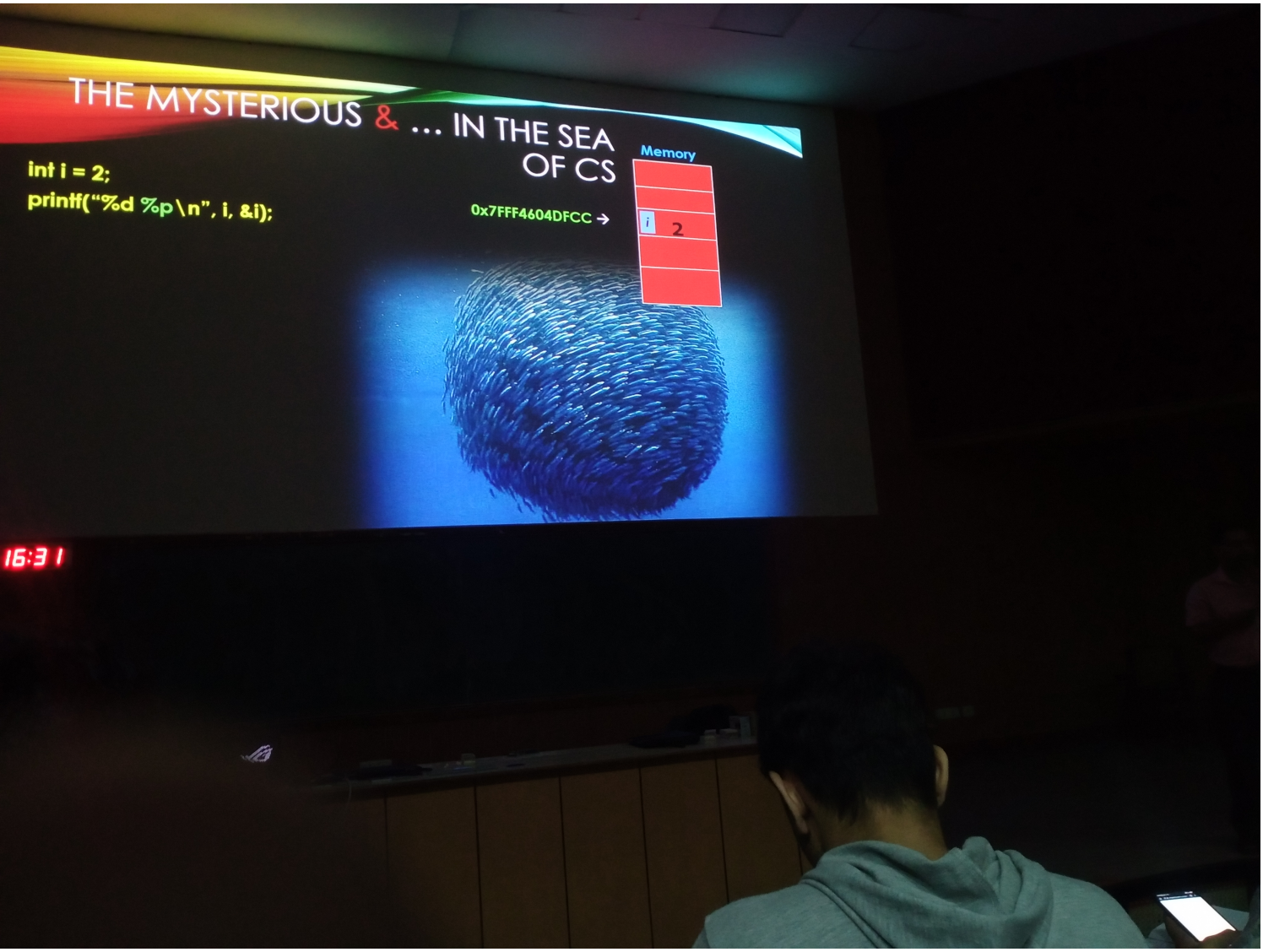

SCANF (FORMATTED INPUT)
int num;
char ch;
scanf("%d", &num); /* reads an integer from stdin into num */
scanf("%c", &ch); /* reads a char from stdin into ch */
16:22

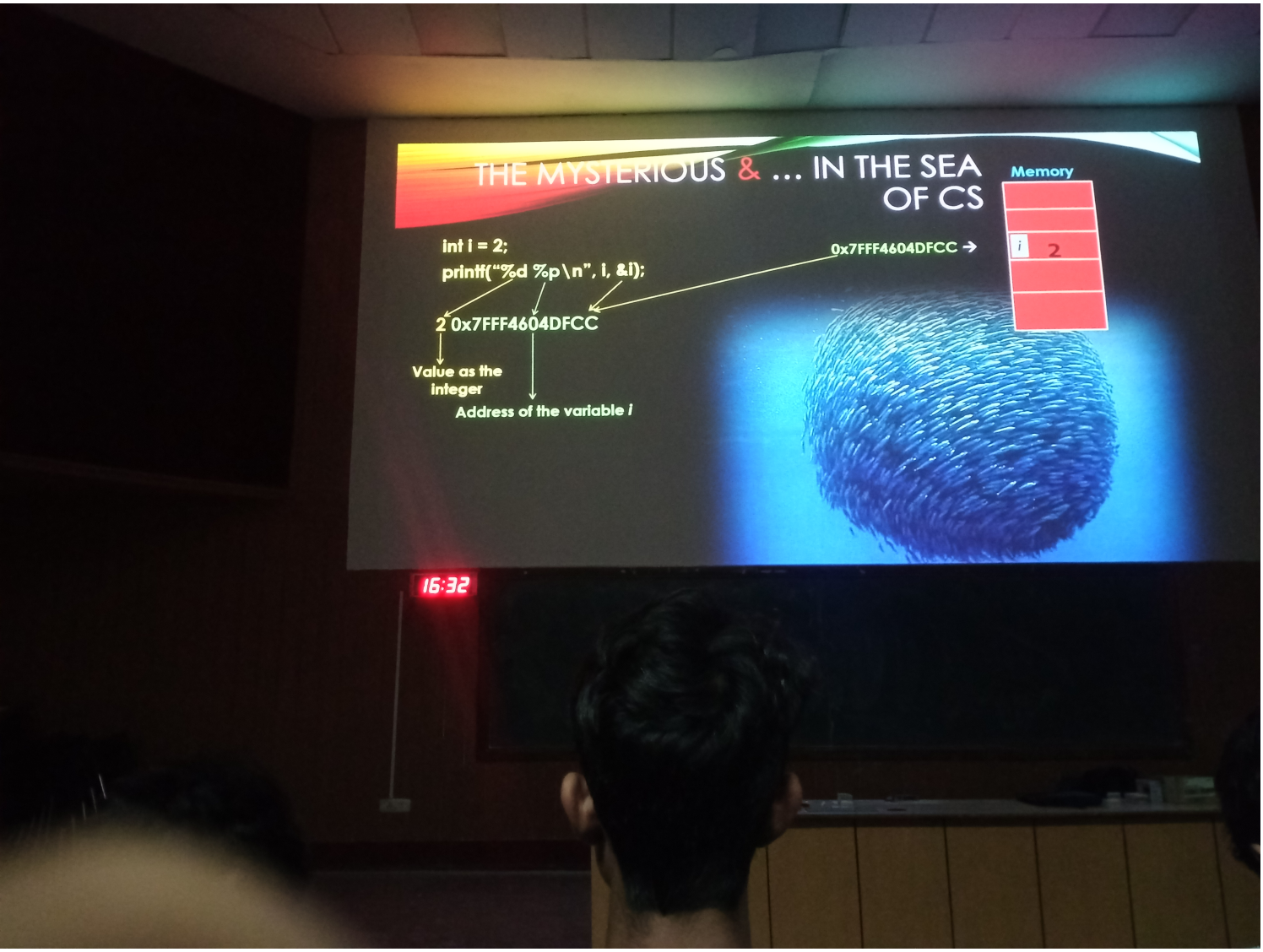

SCANF (FORMATTED INPUT)
/* If multiple inputs need to be read using a single scanf */
scanf("%d %c", &i, &ch);
Format specifiers
int variable
char variable
16:25

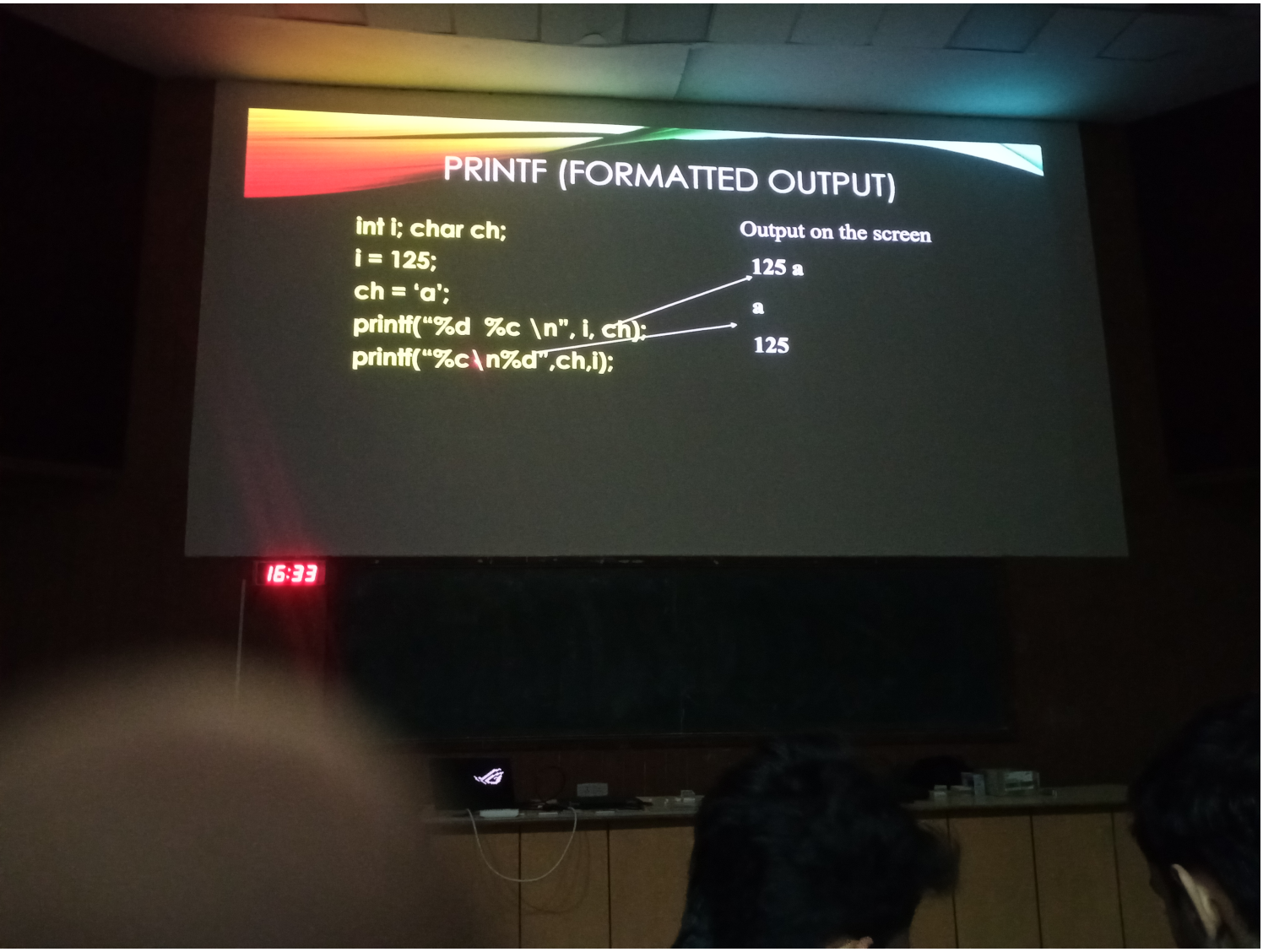

THE MYSTERIOUS '&' IN THE SEA OF CS
• Every variable should have a location in the memory.
• If max_num is the name of a variable then &max_num is its address in the memory.
16:28


THE MYSTERIOUS '&' IN THE SEA OF CS
• Every variable should have a location in the memory.
• If max_num is the name of a variable then &max_num is its address in the memory.
• scanf should be given the addresses of the locations where it should store the values which are being read.
16:28


THE MYSTERIOUS & ... IN THE SEA OF CS
int i = 2;
printf("%d %p\n", i, &i);
16:30


THE MYSTERIOUS & … IN THE SEA OF CS
int i = 2;
printf("%d %p\n", i, &i);
0x7FFF4604DFCC →
Memory
i
2
16:31


THE MYSTERIOUS & … IN THE SEA OF CS
int i = 2;
printf("%d %p\n", i, &i);
2 0x7FFF4604DFCC
Value as the integer
Address of the variable i
0x7FFF4604DFCC
Memory
i
2
16:32


PRINTF (FORMATTED OUTPUT)
int i; char ch;
i = 125;
ch = 'a';
printf("%d %c \n", i, ch);
printf("%c\n%d",ch,i);
Output on the screen
125 a
a
125
16:33

PRINTF

```
char ch = 'A';
printf("%d", ch);
```

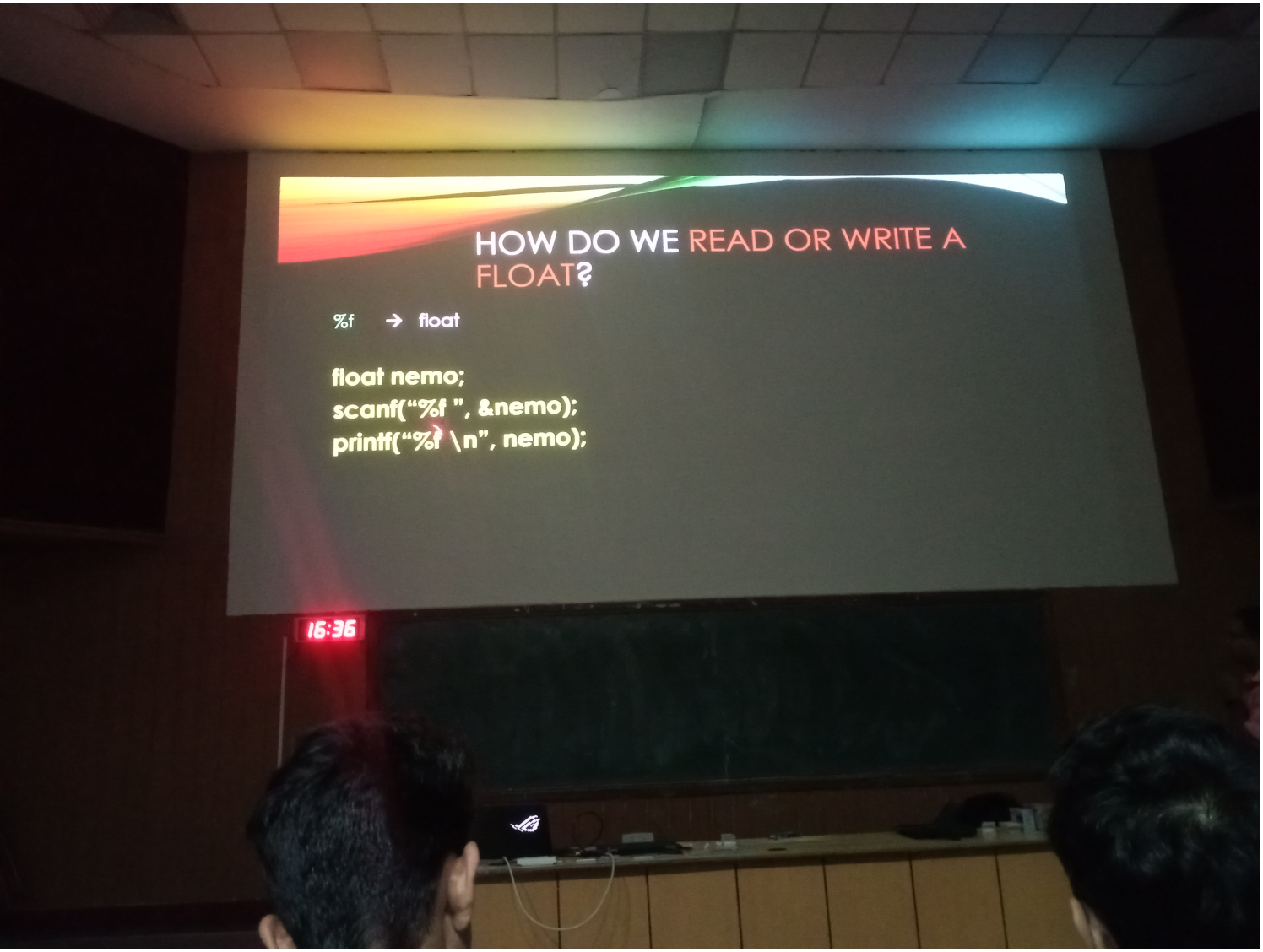
The fish whose names begin with C within the Sea of Cs are proficient in C.
So, let's ask the C... fi... ...tput and why?

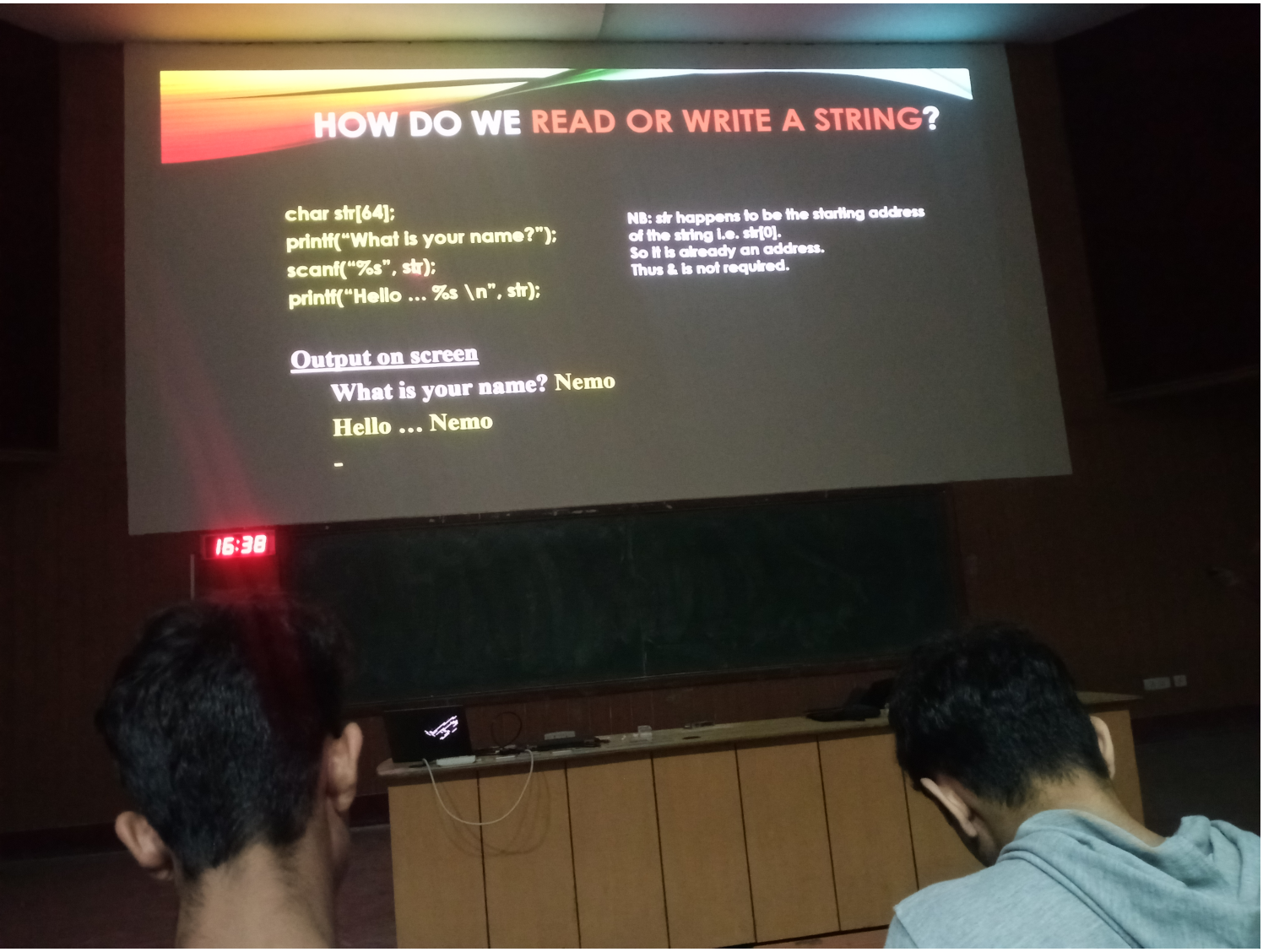
Aha! So you thought you could trick me, eh?
It will print **65** because you used %d
to print the content of a character variable.
NB: ASCII equivalent of A is 65

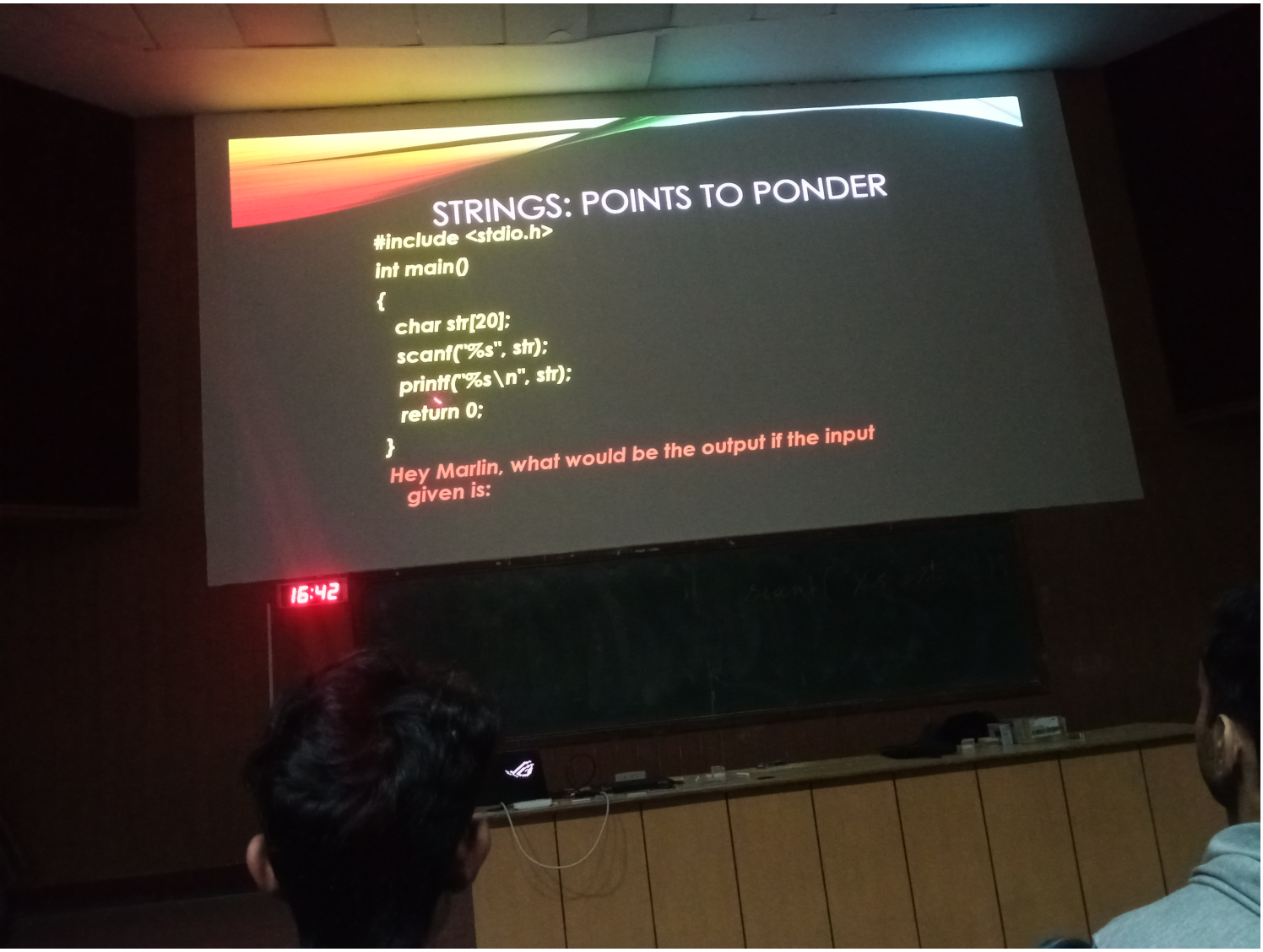
Recognize me?

Marlin from the Great Barrier Reef.
My son, Nemo, got lost when he ventured
into the open C. (Finding Nemo!)

16:35


HOW DO WE READ OR WRITE A FLOAT?
%f → float
float nemo;
scanf("%f ", &nemo);
printf("%f \n", nemo);
16:36


HOW DO WE READ OR WRITE A STRING?
char str[64];
printf("What is your name?");
scanf("%s", str);
printf("Hello ... %s \n", str);
NB: str happens to be the starting address of the string i.e. str[0].
So it is already an address.
Thus & is not required.
Output on screen
What is your name? Nemo
Hello ... Nemo
-
16:38


STRINGS: POINTS TO PONDER
#include <stdio.h>
int main()
{
char str[20];
scanf("%s", str);
printf("%s\n", str);
return 0;
}
Hey Marlin, what would be the output if the input given is:
16:42