

Problem Solving Through Programming in C

Tutorial Session 12

Prof. Anupam Basu
Dept. of Computer Science & Engg.
IIT Kharagpur

Siddhant Mohapatra
PMRF Scholar
IIT Madras

Miscellaneous

Q. What will be output?

```
#include <stdio.h>
int fun(int arr[]) {
    arr = arr + 1;
    printf("%d ", arr[0]);
}
int main(void) {
    int arr[3] = {5, 10, 15};
    fun(arr);
    printf("%d ", arr[0]);
    printf("%d ", arr[1]);
    return 0;
}
```

- a) 5 10 10
- b) 10 5 10
- c) 10 5 15
- d) 10 15 5

arr → arr
 arr → 10
 arr → 15

arr[0] = *(arr + 0)
 *(arr + 0)
 *(arr + 1)

Q. What is the output of the following code?

```
#include <stdio.h>
int main() {
    char *str = "Hello, world!";
    char *ptr = str;
    printf("%s\n", ptr + 7);
    return 0;
}
```

world!

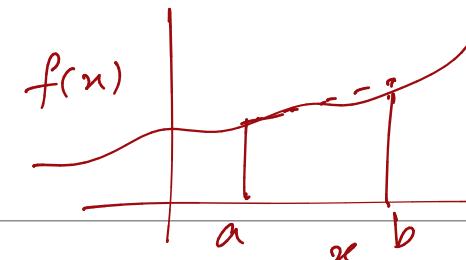
- a) Hello!
- b) world
- c) world!
- d) Hello, world!

ptr → H str
 ptr + 1 → e
 I
 char
 O
 ,
 ptr + 6 → l
 ptr + 7 → o
 r
 I
 d
 !
 \0

Miscellaneous

unsigned → true or 0

(default) signed → true or false or 0



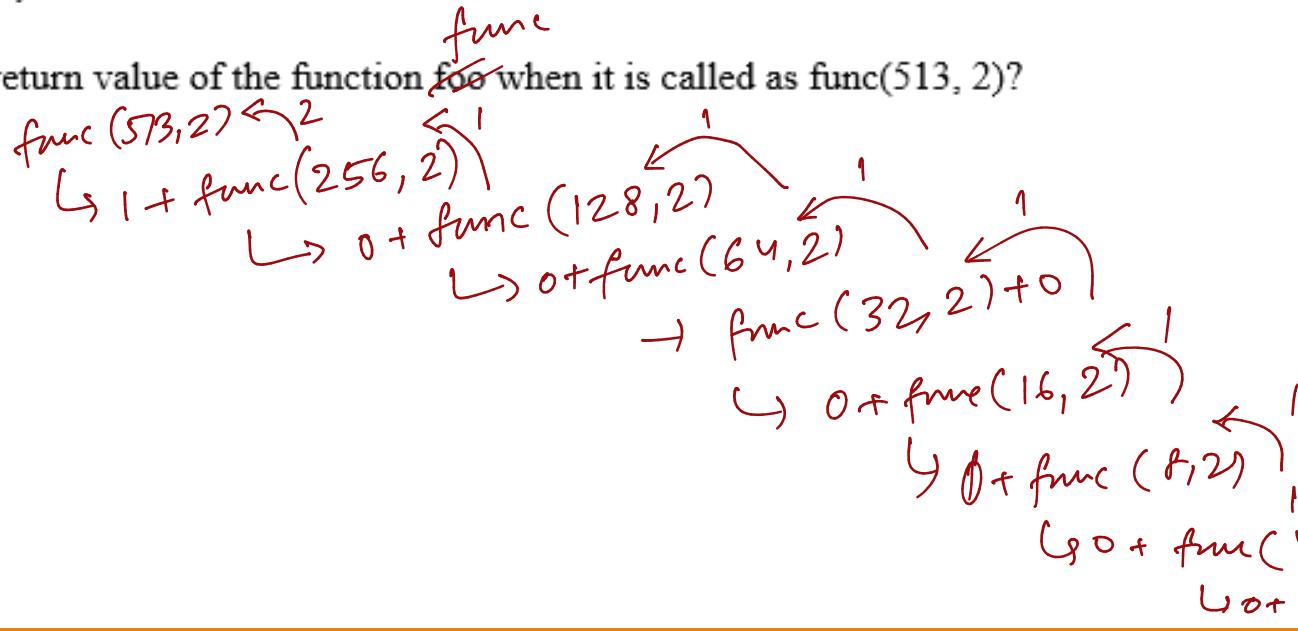
Q.

Consider the same recursive C function that takes two arguments

```
unsigned int func(unsigned int n, unsigned int r)
{
    if (n > 0) return (n%r + func (n/r, r));
    else return 0;
}
```

What is the return value of the function ~~func~~ when it is called as func(513, 2)?

- a) 9
- b) 8
- c) 5
- d) 2



$$a = 0$$

$$b = 1.5$$

$$f(x) = xe^{2x}$$

$$f(a) = 0$$

$$f(b) = 1.5e^{3.0}$$

$$\text{Area} = \frac{1}{2} \times (1.5 - 0) \times (0 + 1.5e^3)$$

$$= 0.75 \times 1.5e^3$$

Miscellaneous

Q. What would be the equivalent pointer expression for referring to the array element $a[i][j][k][l]$?

$\begin{matrix} = & \downarrow & \downarrow & \downarrow & \downarrow \\ & 1st & 2nd & 3rd & 4th \\ \downarrow & \text{dim} & \text{dim} & \text{dim} & \text{dim} \end{matrix}$

- a) $(((*(\text{a}+\text{i})+\text{j})+\text{k})+\text{l})$
- b) $*(*(*(\text{a}+\text{i})+\text{j})+\text{k})+\text{l})$
- c) $(*(*(\text{a}+\text{i})+\text{j})+\text{k}+1)$
- d) $*((\text{a}+\text{i})+\text{j}+\text{k}+1)$

$a \rightarrow \&a[0][0][0][0]$

$*(*(*(\text{a}+\text{i})+\text{j})+\text{k})+\text{l})$

$$\text{num} = \begin{bmatrix} 2 & 5 \\ 11 & 17 \\ 23 & 28 \end{bmatrix}$$

2 ← num
 5
 11
 17
 23 row major format
 28 in C

Q. What will be output when you will execute the following C code?

#include<stdio.h>

int main()

{

short num[3][2]={2,5,11,17,23,28};

printf("%d,%d",*(num+2)[0],**(num+1));

return 0;

}

num

$*(*(\text{num}+2)+0)$

$\text{num}[2][0]$

$**(\text{num}+1)$

$= **(\text{num}+1)+0 = \text{num}[1][0]$

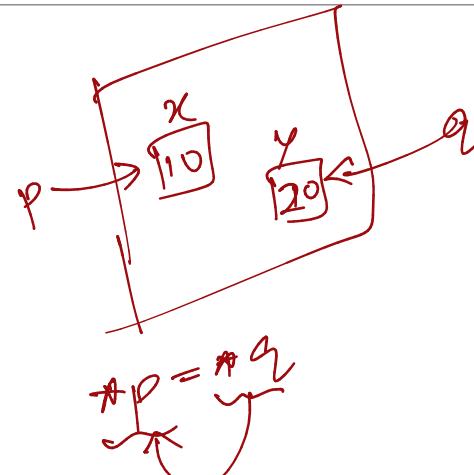
- a) 23, 11
- b) 23, 23
- c) 11, 17
- d) 17, 17

Miscellaneous

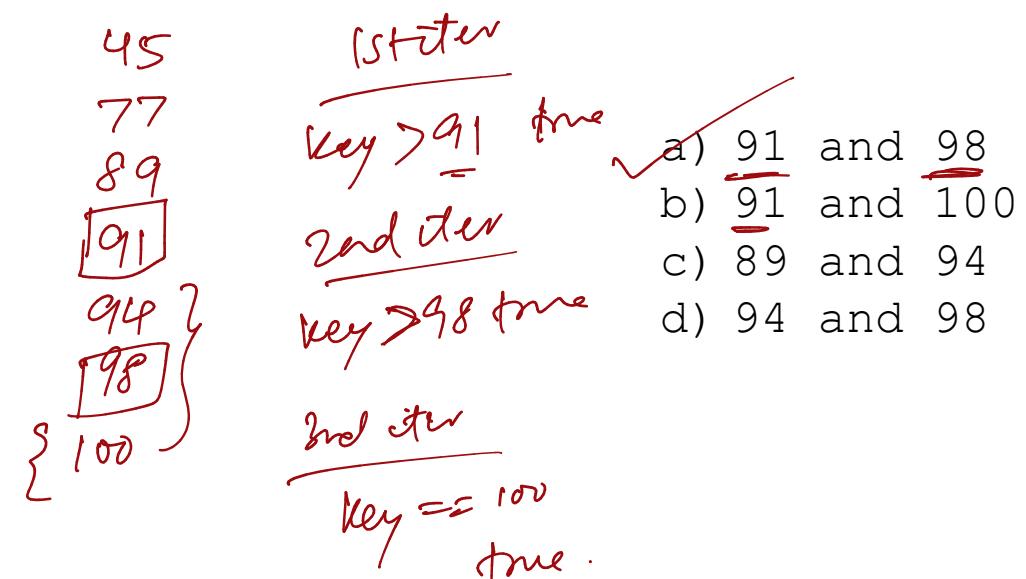
Q. What will be the output?

```
#include<stdio.h>
int main()
{
    int x = 5, y = 10;
    int *p = &x, *q = &y;
    *p = *q;
    *q = 20;
    printf("%d %d", x, y);
    return 0;
}
```

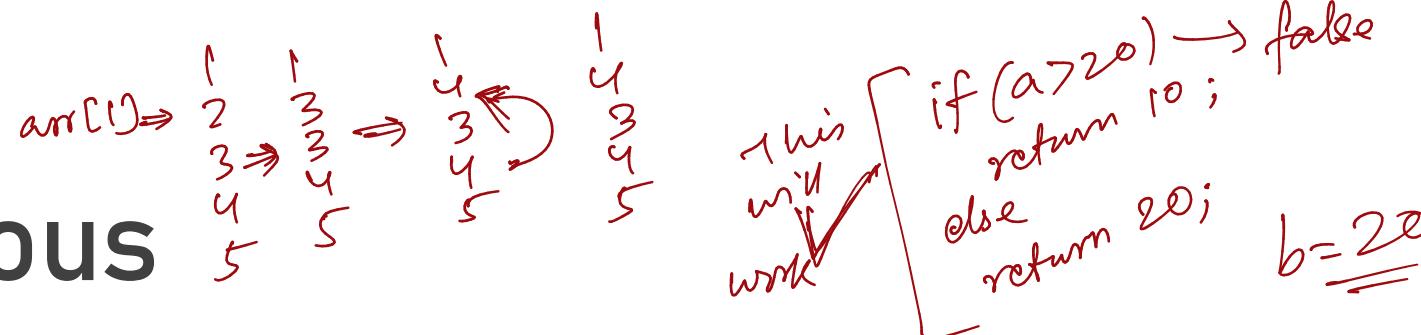
- a) 5 10
- b) 10 20
- c) 20 10
- d) Compilation error



Q. Given an array arr = {45, 77, 89, 91, 94, 98, 100} and key = 100; what are the mid values (corresponding array elements) generated in the first and second iterations?



Miscellaneous



Q. Find the output of the following C program

```
#include<stdio.h>
int main()
{
    int a;
    int arr[5] = {1, 2, 3, 4, 5};
    arr[1] = ++arr[1];
    a = arr[1]++;
    arr[1] = arr[a++];
    printf("%d, %d", a, arr[1]);
    return 0;
}
```

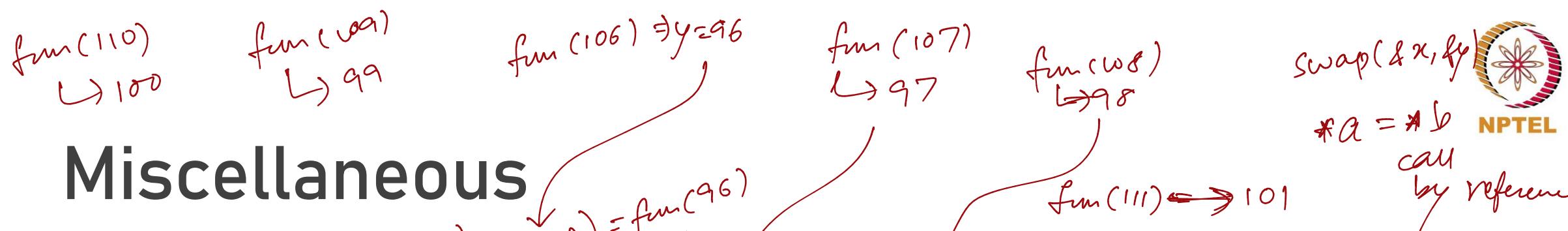
- a) 5, 4
- b) 5, 5
- c) 4, 4
- d) 3, 4

Q. What is the error in the following program

```
#include<stdio.h>
int f(int a)
{ cond if else
    a > 20? return(10): return(20);
}
int main()
{
    int b;
    b = f(20);
    printf("%d\n", b);
    return 0;
}
```

ternary operator \Rightarrow operates on 3 operands

- a) Error: Return statement cannot be used with conditional operators
- b) Error: Prototype declaration
- c) Error: Two return statements cannot be used in any function
- d) No error



Miscellaneous

Q. Consider the function

```
int fun(int x)
{
    int y;
    if(x > 100)
        y = x-10;
    else
        y = fun(fun(x + 11));
    return y;
}
```

For the input x = 95, the function will return

- a) 89
- b) 90
- c) 91
- d) 92

$$\begin{aligned}
 & \text{y} = \text{fun}(\text{fun}(106)) = \text{fun}(96) \\
 & \text{y} = \text{fun}(\text{fun}(107)) = \text{fun}(97) \\
 & \text{y} = \text{fun}(\text{fun}(108)) = \text{fun}(98) \\
 & \text{y} = \text{fun}(\text{fun}(109)) = \text{fun}(99) \\
 & \text{y} = \text{fun}(\text{fun}(110)) = \text{fun}(100) \\
 & \text{y} = \text{fun}(\text{fun}(111)) = \text{fun}(101) = 91
 \end{aligned}$$

Q. What is the output of the following code snippet?

```
#include <stdio.h>
void swap(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
}
call by value
```

```
int main() {
    int x = 1, y = 2;
    swap(x, y);
    printf("x = %d, y = %d\n", x, y);
    return 0;
}
```

- a) x=1, y=2
- b) x=2, y=1
- c) Compilation error
- d) Runtime error

Miscellaneous

Q. What will be the output?

```
#include <stdio.h>
int main()
{
    int a[2][3] = {1, 2, 3, 4};
    int i = 0, j = 0;
    for (i = 0; i < 2; i++)
        for (j = 2; j >= 0; j--)
            printf("%d", a[i][j]);
    return 0;
}
```

321004

*a = [1 2 3]
 | 0 0] 2x3*

*i = 0, j = 2 a[0][2]
 i = 0, j = 1 a[0][1]
 i = 0, j = 0 a[0][0]
 i = 1, j = 2 a[1][2]
 i = 1, j = 1 a[1][1]
 i = 1, j = 0 a[1][0]*

1 compound statement

*if (any other
 than 0)
 true -
 if(0) false*

*a → '0'
 still counts
 as 1 character*

Q. What will be the output?

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

```
int main()
```

```
{
```

```
int i;
```

```
char a[] = "";
```

```
if (printf("%s", a))
```

```
    printf("The string is empty");
```

```
else
```

```
    printf("The string is not empty");
```

```
return 0;
```

```
}
```

- if (1) ← returns phno. of
 true ← characters being
 sent to be printed*
- a) The string is empty
 - b) The string is not empty
 - c) Compilation error
 - d) None of the above

$$\text{ary} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$A = \begin{bmatrix} 5 \\ 4 \\ 9 \\ 1 \\ 3 \end{bmatrix}$$

Miscellaneous

Q. What is the output of the following C code?

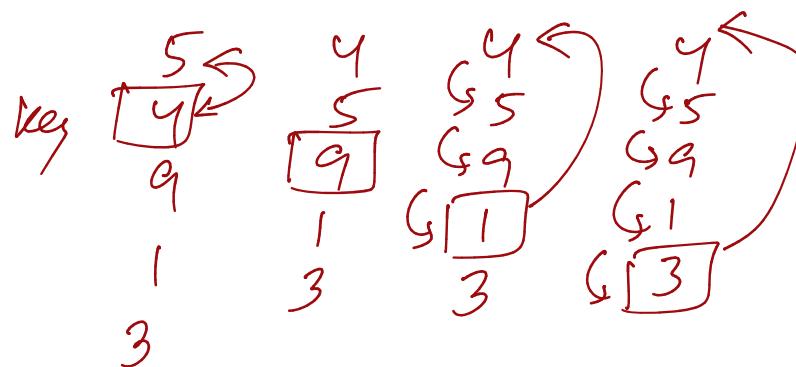
```
#include <stdio.h>
int main()
{
    int ary[3][3];
    ary[][] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    printf("%d\n", ary[2][0]);
    return 0;
}
```

*arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
arr[3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};*

2D array initialization

- a) Compilation error
 - b) 7
 - c) 1
 - d) 2
- Always mention the size of 2nd dim*

Consider the array $A[] = \{5, 4, 9, 1, 3\}$ apply the insertion sort to sort the array. Consider the cost associated with each sort is 25 rupees, what is the total cost of the insertion sort when element 1 reaches the first position of the array?



- a) 25
- b) 50
- c) 75
- d) 100

total of 4 steps reqd. for 1 to reach 1st element of array

3 steps reqd. for 1 to reach 1st element of array

$$= 25 \times 3$$

Miscellaneous

- Q. A one-dimensional array X has indices 1....50. Each element is an int. The array is stored at location 1000 decimal. The starting address of X[27] is 1108 (assume int takes 4 bytes).

