

GATE 2015 – C Programming | Question 21

Recursion, Call Stack and LIFO

Problem

Consider the following C function. What is its output for the input "ABCD EFGH"?

```
void foo(char *a)
{
    if (*a && *a != ' ')
    {
        foo(a + 1);
        putchar(*a);
    }
}
```

(A) ABCD EFGH (B) ABCD (C) HGFE DCBA (D) DCBA

Key Observation

The function moves forward with `foo(a + 1)`, but `putchar(*a)` is placed after the recursive call. Therefore, characters are printed while the recursive calls return. The recursion stops at the first space.

1. Going Deeper

For the input "ABCD EFGH", the pointer moves:

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow \text{SPACE}$

At A, B, C and D the condition is true, so another recursive call is made. When a points to the space, `*a != ' '` becomes false and recursion stops.

2. Call Stack

The recursive calls are stored in the function call stack:

```
foo("ABCD EFGH")
  foo("BCD EFGH")
    foo("CD EFGH")
      foo("D EFGH")
        foo(" EFGH") ← stops
```

The last call is at the top of the stack.

3. Unwinding / Returning

Now the calls return in LIFO (Last In, First Out) order. During each return, `putchar(*a)` executes:

$D \rightarrow C \rightarrow B \rightarrow A$

The space is not printed because its call never enters the if block.

Final Output

DCBA → Correct Answer: (D)

Remember

foo(a + 1) → goes forward through the string.

putchar(*a) → executes while coming back.

First space → stops recursion.

Therefore → characters before the space are printed in reverse order.

Easy pattern: GO FORWARD: $A \rightarrow B \rightarrow C \rightarrow D$ | COME BACK: $D \rightarrow C \rightarrow B \rightarrow A$

Complete Program for Testing

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

```
void foo(char *a)
{
    if (*a && *a != '\0')
    {
        foo(a + 1);
        putchar(*a);
    }
}
```

```
int main()
{
    char str[100];

    scanf("%99[^\n]", str);

    foo(str);

    return 0;
}
```

Sample input:
ABCD EFGH

Output:
DCBA

Code
Learn
Practice
Improve
Succeed

Thank You for Reading!

Keep Learning | Keep Practicing | Keep Growing

C PROGRAMMING
DATA STRUCTURES
ALGORITHMS
GATE PREPARATION

Good Code
Brighter Future

Understand Concepts
Solve Problems
Improve Step by Step
Stay Consistent
Crack GATE

"A small step every day leads to big results."

You Can Do It!

Happy Learning... />

EngineersTutor.com
Your Learning Partner for Programming & GATE

EngineersTutor
Subscribe for More Tutorials

C Programming Data Structures Algorithms GATE Preparation