What Happens Next?

```
int32_t print_reverse(const char* s)
{
    int32_t rest;
    if ('\0' == *s) {
        return 0;
    }
    rest = print_reverse (s + 1);
    printf ("%c", *s);
    return (rest + 1);
}
```

Output is now "**w**".

Return (**rest + 1**).

Return (**rest + 1**)

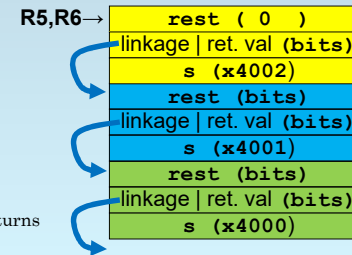What is **rest**?   **0**

Adding 1, we obtain 1.

call depth 1
p_r ("Now")

call depth 2
p_r ("ow")

R5,R6→

| rest (0) |
| linkage \| ret. val **(bits)** |
| s **(x4002)** |
| rest **(bits)** |
| linkage \| ret. val **(bits)** |
| s **(x4001)** |
| rest **(bits)** |
| linkage \| ret. val **(bits)** |
| s **(x4000)** |

call depth 3
p_r ("w")

call depth 4
p_r ("")   returns 0

Write 1 into Return Value Slot

call depth 1
p_r ("Now")

call depth 2
p_r ("ow")   R5,R6→

| rest (0) |
| linkage \| ret. val (**1**) |
| s **(x4002)** |
| rest **(bits)** |
| linkage \| ret. val **(bits)** |
| s **(x4001)** |
| rest **(bits)** |
| linkage \| ret. val **(bits)** |
| s **(x4000)** |

call depth 3
p_r ("w")   returns 1

call depth 4
p_r ("")   returns 0
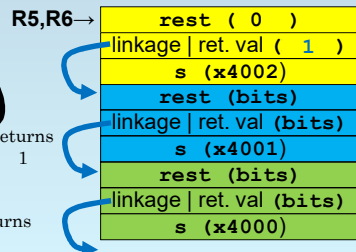
What Happens When **print_reverse** Returns?

```
int32_t print_reverse(const char* s)
{
    int32_t rest;
    if ('\0' == *s) {
        return 0;
    }
    rest = print_reverse (s + 1);
    printf ("%c", *s);
    return (rest + 1);
}
```

On return, return value is written into **rest**.