Quiz 5 Version B

Due No due date   **Points** 8   **Questions** 3   **Available** after Nov 11 at 1pm
Time Limit None   **Allowed Attempts** Unlimited

Instructions

Quiz for Lecture 5: Recursion

7 points required to pass

<div align="center">

Take the Quiz Again

</div>

Attempt History

	Attempt	Time	Score
LATEST	**Attempt 1**	less than 1 minute	0 out of 8 *

<div align="center">* Some questions not yet graded</div>

Score for this attempt: **0** out of 8 *
Submitted Nov 28 at 2:28pm
This attempt took less than 1 minute.

Question 1	**Not yet graded / 2 pts**

The sum of integers from 1 through N is simply:

1 + 2 + 3 + ... + N

This can be defined recursively as:

Sum(1) == 1

Sum(N) = Sum(N - 1) + N, for N > 0

Write a *recursive* implementation of this sum of integers function named

Sum:

unsigned Sum(unsigned N)

{

   // your code goes here

}

Your Answer:

if (N == 1) return 1;

return Sum(N - 1) + N;

Question 2   **Not yet graded / 2 pts**

Write a *recursive* implementation of the function **put_str_rev**, that takes a C-string as its argument, and that displays the characters in the C-string to the screen *in reverse order*.

For example, this code in **main**:

char a[10] = "hello";

put_str_rev(a);

would display:

olleh

void put_str_rev(const char *s)

```
{
    // your code goes here
}
```

Your Answer:

```
if (*s) {
    put_str_rev(s + 1);
    cout << *s;
}
```

Question 3   **Not yet graded / 4 pts**

Define a function named **put_int_bits** that takes an **int** value as its argument, and that displays the 32-bit binary representation of that **int** value to **cout**:

0   00000000000000000000000000000000

7   00000000000000000000000000000111

13   00000000000000000000000000001101

-1   11111111111111111111111111111111

put_int_bits returns no value. It may be useful to have **put_int_bits** call one or more helper functions. At least one function in your implementation should be a *recursive* function.

Your Answer:

fdsa

```cpp
void put_int_bit_n(unsigned i, unsigned n)
{
    if (n) {
        put_int_bit_n(i / 2, n - 1);
        cout << i % 2;
    }
}
void put_int_bits(int i)
{
    put_int_bit_n(i, 32);
}
```

Quiz Score: **0** out of 8