Quiz 5 Version F

Due No due date   **Points** 8   **Questions** 3
Available after Nov 21 at 3:30pm   **Time Limit** None
Allowed Attempts Unlimited

Instructions

Quiz for Lecture 5: Recursion

7 points required to pass

Take the Quiz Again

Attempt History

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

Some questions not yet graded

Score for this attempt: **0** out of 8 *
Submitted Nov 28 at 2:30pm
This attempt took less than 1 minute.

Question 1	**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:

fdas

```
if (*s) {

    put_str_rev(s + 1);

    cout << *s;

}
```

Question 2   **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

-7   11111111111111111111111111111001

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:

```
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);

}
```

Question 3   **Not yet graded / 2 pts**

N! (**N** factorial) can be defined iteratively as:

0! = 1

N! = 1 * 2 * 3 * ... * (N-1) * N, for **N > 0**

or recursively as:

0! = 1

N! = N * (N-1)!, for **N > 0**

Write a *recursive implementation* of the function **factorial**, that takes a non-negative integer as its argument **N** and returns **N!**

unsigned factorial(unsigned N)

{

    // this is the code you have to write

}

Your Answer:

```
if (N == 0)

    return 1;

return N * factorial(N - 1);
```

Quiz Score: **0** out of 8