# Quiz 5 Version D

**Due** No due date **Points** 8 **Questions** 3
**Available** after Nov 16 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 2:29pm
This attempt took less than 1 minute.

### Question 1 — Not yet graded / 3 pts

Suppose we have this structure definition, for a node in a singly linked list of **char**:

**struct cl_node { char data; cl_node *next; };**

Write ***a recursive implementation*** of this function, that displays the **char** values in the argument list to **cout** in order. ***(An iterative implementation is not acceptable.)*** For example, if the list of characters

contains **'a'**, **'b'**, **'c'**, the function should display **abc** to **cout**. If the list is empty, the function should do nothing.

**void clist_display(const cl_node *p)**

**{**

// this is the part you have to write as your answer

**}**

Your Answer:

fdsa

```
if (p == nullptr)
    return;
cout << p->data;
 clist_display(p->next);
```

## Question 2

**Not yet graded / 2 pts**

Write a recursive implementation of this function, that displays **N** alternating **0**s and **1**s to **cout**, where the first digit displayed is **0** if **N** is odd and **1** if **N** is even. For example, **alter_0_1(3)** should display **010**, and **alter_0_1(8)** should display **10101010**. If **N** is **0**, the function should do nothing.

**void alter_0_1(unsigned N)**

**{**

// this is the part you have to write as your answer

**}**

Your Answer:

```
if (N == 0)
    return;
if (N % 2)      // N is odd
    cout <<  0;
else
    cout << 1;
alter_0_1(N - 1);
```

## Question 3

**Not yet graded / 3 pts**

Suppose we have this structure definition, for a node in a singly linked list of **double**:

**struct dl_node {  double data;  dl_node *next;  };**

Write ***a recursive implementation*** of this function, that returns the sum of the values in the linked list. ***(An iterative implementation is not acceptable.)*** For example, if the list of **double** contains **1.1**, **2.2**, **3.3**, the function should return **6.6**. If the list is empty, the function should return **0.0**.

```
double dlist_sum(const dl_node *p)
{
    // this is the part you have to write as your answer
```

```
}
```

Your Answer:

```
if (p == nullptr)
    return 0.0;
return p->data + dlist_sum(p->next);
```

Quiz Score: **0** out of 8