

# Quiz 3



# Sample

This quiz consists of 1 question. You have 7 minutes.

Assume the ArmLab/gcc217 environment unless otherwise stated in a problem.

Do all of your work on these pages. This quiz is preprocessed by computer, so for your final answers to be scored you must write them inside the designated spaces. Please make text answers dark and neat.

Name:  NetID:

This is a closed-book, closed-note quiz. Please place items that you will not need out of view in your bag or under your working space at this time. Electronic devices such as cell phones, laptops, smartwatches except to check the time, etc. may not be used during this quiz.

This quiz is administered under the Princeton University Honor Code. All suspected violations of the Honor Code must be reported to [honor@princeton.edu](mailto:honor@princeton.edu).

In the box below, copy **and** sign the Honor Code pledge before turning in your quiz:  
*"I pledge my honor that I have not violated the Honor Code during this examination."*

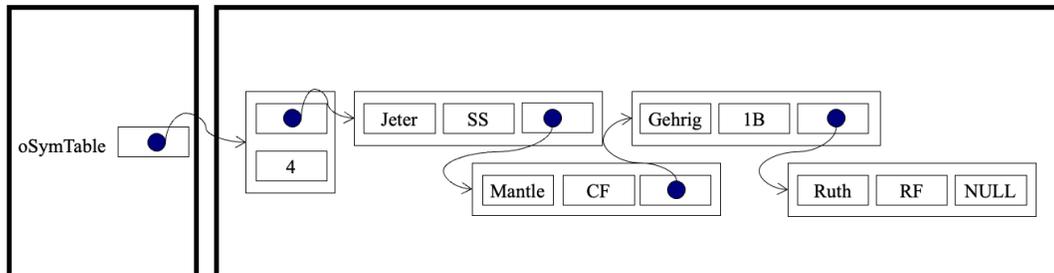
X \_\_\_\_\_

The question is on the back of this page. Do not turn this page over until instructed to begin.

## Question 1: Symbol Table Function Implementation (7 points)

An A3 linked list helper function `SymTable_prevBinding` takes two parameters: a pointer to an instance of a symbol table and a string that represents a key. It returns a pointer to the node in the list that *comes before* the node with that key, if there is one, and `NULL` otherwise.

For example, a call with the key argument `"Gehrig"` should return a pointer to the node with key `"Mantle"`, but calls with the keys `"Jeter"` or `"Judge"` should return `NULL`:



In your code below, assume that the linked list structures and types are defined as follows:

```
typedef struct Node *Node_T;
struct Node {
    char *pcKey;
    void *pvValue;
    Node_T psNext;
};

#include <stddef.h>
typedef struct ST *SymTable_T;
struct ST {
    Node_T psFirst;
    size_t ulCount;
};
```

Write the complete function definition for `SymTable_prevBinding` in the box below, using the given structure types and member names. You may omit parameter validation `asserts`.

```
#include <string.h>
static Node_T SymTable_prevBinding(SymTable_T oST, const char *pcKey) {
```

```
}
```