Quiz1

● Graded

Student

Krisnajit Rajeshkhanna

Total Points

26 / 26 pts

Question 1

delete_items_by_name

25 / 25 pts

✓ **– 0 pts** does not (or incorrectly) free node **name** ; *higher penalty could be assigned for incorrect free for later tests*

Question 2

write name on p3

1 / 1 pt

✓ **+ 1 pt** wrote name on p3

CMSC216 Quiz 1
Fall 2023
October 24

First name (PRINT): Krisnajit

Last name (PRINT): RajeshKhanna

Numeric University Id: 119769843

I pledge on my honor that I have not given or received any unauthorized assistance on this test.

Your Signature: R.

- This is a **15-min**, closed-textbook, closed-slides, and closed-notes quiz.
- **Total point value is 25** and will be mapped to 2.5% of your course grade.
- **Put your name inside the box located at the top right corner of Page 3. You will get 1 extra point.** The extra point can only be used to make up your points. The maximum points for this test will be capped at 25.
- **WRITE NEATLY**. If your answer is too hard to read, we will not grade it (i.e., 0 point).
- In the questions, _ symbol represents a space character.
- Assume that the compilation options for this course are used for the questions.
- If the execution result of a code snippet is platform-dependent, assume that code is run on the `grace` machine.

DO NOT SHARE THE QUESTIONS ON-/OFF-LINE.

Q1. Linked List

Consider the following struct defined as PartyItem and a linked list of PartyItems.

```
typedef struct party_item {
  unsigned int price;
  unsigned int qty;
  char *name;
  struct party_item *next;
} PartyItem;
```

Write a function named delete_items_by_name that deletes PartyItems from the list if their names are equal to the name given as a function parameter. The function prototype is provided below:

```
delete_items_by_name(PartyItem** head, const char* name)
```

Note that the first parameter is the pointer to the pointer to the head node (a PartyItem with valid field values) (i.e., we assume that the list does not use a dummy node.)

The following list provides additional requirements and assumptions that should be considered:

(a) You do not need to give #include lines.

(b) You are allowed to use the string library functions (e.g., strcpy, strcmp)

(c) The linked list should be correctly updated after deletion(s).

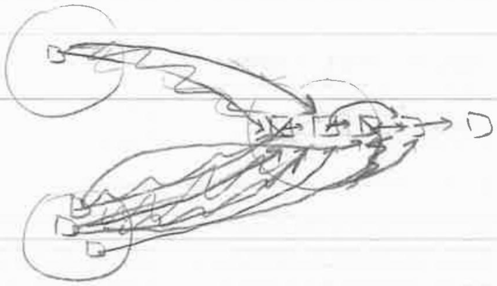
An **example** driver code is given below. Note that the code assumes that the following items have been added to the list in the order: balloon, decoration, food, games, balloon. In the example, the delete_items_by_name function will delete 2 nodes from the list (first and the last) and will make the head in main point to the second node (decoration) in the original list.

```
int main() {
  PartyItem *head = NULL;
  int count;

  /*
    Assume that code to add the following items is given here:
    balloon, decorations, food, games, balloon
   */

  delete_items_by_name(&head, "balloon");

  /* Assume that more code is given here */

  return 0;
}
```

Give your code on the next page.

Krisnajit Rajeshkhanna

```
void delete_items_by_name(PartyItem **head, const char *name) {
 PartyItem *curr = *head, *prev = NULL; *dummy;
 while (curr != NULL) {
     if ( strcmp(curr->name, name) ==0) {
          if(prev == NULL) {
               *head = curr->next;
          }
          else {
             prev->next=curr->next;
          }
          dummy = curr;
          curr = curr->next;
          free (dummy);
      }
      else {
          prev = curr;
          curr = curr->next;
      }
  }
}
```



This page is left blank intentionally.