Structure definition for Worksheet 8:

```
struct pizzaMaker
{
   char topping [30];          // the topping you want
   struct pizzaMaker * next;  //pointer to next topping (if you want more toppings)
};

typedef struct pizzaMaker pizzaNode;
```

Q 1: Create a pointer called headPizza that will point to the start of a list (head node), once the list is created. Since the list currently is empty, make headPizza point to NULL for now.

```
pizzaNode *headPizza = NULL;
```

Q 2: Create a new node called pnode1 of type pointer to pizzaNode. Set *topping* of this new node to "Onion" and *next* to NULL. Make this the head node.

```
pizzaNode *pnode1;
pnode1 = malloc(sizeof(pizzaNode));
strcpy(pnode1 -> topping, "Onion");
pnode1 -> next = NULL;
headPizza = pnode1;
```

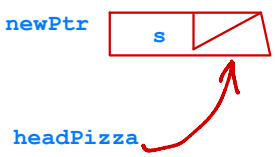
Q 3: Assume that you have the following Linked list:

Write code to create a new node called newNode (where newNode is of type pizzaNode *). Prompt the user for a pizza topping of their choice to set the value of *topping* of newNode. Add this node as the last node of the list. So if the user enters Spinach, the new list now becomes:

Q 4: What does the following function do to the list given above in Q3, if the function is called as
headPizza = fun ("Pepperoni", headPizza);

```
Q3)
pizzaNode *newNode;
newNode = malloc(sizeof(pizzaNode));
scanf("%s", newNode -> topping);
newNode -> next = NULL;
```

```
181 pizzaNode * fun (char *s, pizzaNode * head)
182 {
183
184     pizzaNode * newPtr = (pizzaNode *)(malloc(sizeof(pizzaNode)));
185
186
187     strcpy(newPtr->topping, s);
188     newPtr->next = NULL;
189
190     if (head == NULL) {
191         head = newPtr;
192     }
193     else {
194         newPtr->next = head;
195     }
196
197     return newPtr;
198 }
```



Q 5: a. Write a function called displayAllToppings that takes the head of the linked list (headPizza) and displays all toppings in the list, each separated by a newline.

b. Write 5a using a recursive function displayAllToppingsR.

```
// Q5a
void displayToppings(pizzaNode *head) {
    while(head != NULL) {
        printf("%s\n", head -> topping);
        head = head -> next;
    }
}

// Q5b
void displayToppingsR(pizzaNode *head) {
    if(head != NULL) {
        printf("%s \n", head -> topping);
        displayToppingsR(head -> next);
    }
}
```