Atividades da semana 9 - RESPOSTAS

Questão 1:
- Print the Elements of a Linked List

obs: Não coloquei o código todo, segue **apenas** a parte faltante para o algoritmo funcionar no HackerHank.

```
void printLinkedList(SinglyLinkedListNode* head) {
    SinglyLinkedListNode* current = head;
    // Percorre a lista ligada e imprime cada elemento
    while (current != NULL) {
        printf("%d\n", current->data);
        current = current->next;
    }
}
```

Questão 2:
- Insert a Node at the Tail of a Linked List

obs: Não coloquei o código todo, segue **apenas** a parte faltante para o algoritmo funcionar no HackerHank.

```
// Função que insere um novo nó no final da lista encadeada
SinglyLinkedListNode* insertNodeAtTail(SinglyLinkedListNode* head,
int data) {
    // cria um novo nó
    SinglyLinkedListNode* newNode =
create_singly_linked_list_node(data);

    // se a lista estiver vazia, o novo nó é o primeiro nó
    if (head == NULL) {
        head = newNode;
    }
    // se a lista não estiver vazia, encontra o último nó e
adiciona o novo nó ao final
    else {
        SinglyLinkedListNode* current = head;
        while (current->next != NULL) {
            current = current->next;
        }
        current->next = newNode;
```

```
    }
    return head;
}
```

Questão 3:
- Insert a node at a specific position in a linked list

obs: Não coloquei o código todo, segue **apenas** a parte faltante para o algoritmo funcionar no HackerHank.

```
SinglyLinkedListNode* insertNodeAtPosition(SinglyLinkedListNode*
llist, int data, int position) {
    SinglyLinkedListNode *new_node =
create_singly_linked_list_node(data);
    SinglyLinkedListNode *current_node = llist;
    int i = 0;

    if (position == 0) {
        new_node->next = llist;
        return new_node;
    }

    while (current_node != NULL) {
        if (i == position-1) {
            new_node->next = current_node->next;
            current_node->next = new_node;
            break;
        }

        current_node = current_node->next;
        i++;
    }

    return llist;
}
```