COMP 2401 B

Test #2 (version 2)

1. [2 marks] d
2. [2 marks] a
3. [2 marks] c
4. [2 marks] a
5. [2 marks] c (alt: b)
6. [2 marks] d

7. [10 marks]

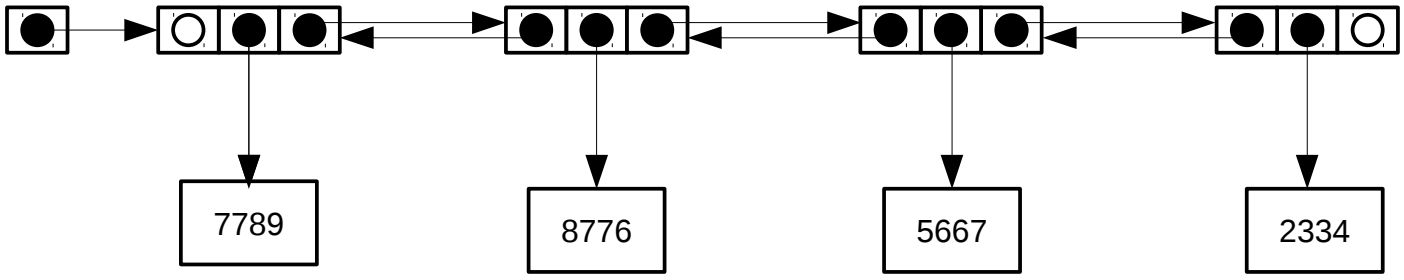
```
void initRunner(char *n, int r, RunnerType **runner) {
  *runner = malloc(sizeof(RunnerType));
  strcpy((*runner)->name, n);
  (*runner)->rank = r;
}

int main()
{
  RunnerType *newRunner;
  initRunner("Gertrude", 3, & newRunner);
  printf("Name is %s, rank is %d\n",newRunner->name,newRunner->rank);
  free(newRunner);
}
```

Marking:
-- 2 marks for making parameter a double pointer in `initRunner()`
-- 2 marks for allocating `RunnerType` in `initRunner()`
-- 2 marks for dereferencing `runner` in `initRunner()` (1 mark each)
-- 2 marks for passing address of `newRunner` to `initRunner()`
-- 2 marks for freeing `newRunner`

8. [28 marks]

a. [6 marks]



Marking:
-- 1 mark for correct pointer to head node
-- 1 mark for first node's prev set to null
-- 1 mark for last node's next set to null
-- 1 mark for 3 next pointers
-- 1 mark for 3 prev pointers
-- 1 mark for correct pointers to data structures, in correct order

b. [10 marks]

```
  NodeType *newNode;

// 4 marks for allocating and initializing node
// -- 2 marks for malloc (zero if freed)
// -- 2 marks for initializing node data and prev
  newNode = (NodeType *) malloc(sizeof(NodeType));
  newNode->data = newAcct;
  newNode->prev = NULL;

// 2 marks for setting new node's next to head
  newNode->next = list->head;

// 2 marks for checking that old head is not null
// and setting old head's prev to new node
  if (list->head != NULL)
    list->head->prev = newNode;

// 2 marks for setting new head
  list->head = newNode;
```

c. [12 marks]

```
  NodeType *newHead;
  AcctType *goner;

// 2 marks for dealing with empty list case
  if (list->head == NULL)
    return 0;

// 2 marks for saving new head
  newHead = list->head->next;

// 1 mark for saving current head's data
  goner = list->head->data;

// 2 marks for freeing current head node
  free(list->head);

// 2 marks for setting new head
  list->head = newHead;

// 2 marks for setting new head's prev to NULL
  newHead->prev = NULL;

// 1 mark for returning last node's data
  return goner;
```