

Test 2C: Programming

Due No due date

Points 15

Questions 1

Available May 3 at 7:15pm - May 3 at 8:30pm about 1 hour

Time Limit 30 Minutes

Instructions

Write the code of the functions based on the given specifications. Full points are given to functions written efficiently. In each of the function, there should only be one (1) return statement ONLY and conditions in the iteration and alternation statements should be a relational expression.

Preformat your code. Highlight code, Under Paragraph press Preformatted.

Example:

```
printf("\nHello World");
```

This quiz was locked May 3 at 8:30pm.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	21 minutes	0 out of 15

❗ Correct answers are hidden.

Score for this quiz: 0 out of 15

Submitted May 3 at 7:41pm

This attempt took 21 minutes.

Question 1

0 / 15 pts

Given the definition:

```
#define MAX 0XF
typedef enum {
    UNIMPORTANT, LATER, IMMEDIATE
}PRIORITY;
typedef struct to_do_task {
    PRIORITY p; /* task priority */
```

```

char desc[50]; /* task description */
int time;      /* task time in minutes */
}TDT;

/* Linked List Representation of List */
typedef struct node{
    TDT task;
    struct node *nextTask;
}*taskLList; /* Definition of the ADT List */

```

Based on the given data structure definition, write the code of the function. There should be no additional data structure (array or list) in your code other than the data given as parameters.

Function Specification. The function will sort in ascending order the given list of tasks implemented using linked list according to the time each task is done. Use the bubble sort technique that pushes tasks with longer time to the end of the linked list. **Note:** Use for-loop whenever an iterative statement is necessary. The order of dynamically allocated spaces will not be affected in the function.

Your Answer:

```

void sortLinkedList(taskLList* start){
    taskLList* var1;
    taskLList* var2=NULL;
    taskLList temp;
    do{
        var1 = start;
    }while (var1 -> next != var2){
        if(var1->task.time > var1->nextTask->task.time){
            temp = (taskLList) malloc(sizeof(struct node));
            temp->task = var1->task;
            var1->nextTask->task = var1->task;
            var1->nextTask->task = temp->task;
        }
        var1 = var1->nextTask;
    }
    var2 = var1;
}

```

The instruction is to use for-loop!!!!

Quiz Score: **0** out of 15