King Saud University
College of Computer and Information Sciences
Computer Science Department

Semester 3: 2022-2023
Lab Exam

Write a program that manages a to-do list of tasks. It should consist of the following files.

todo.h

```
/*directives/header guards*/

typedef enum { ONGOING, COMPLETED
} Status;

typedef struct {
/* a description of the task as a
dynamically allocated string. */
/* the current status of the task. */

} Task;

typedef struct {
/* a dynamically allocated array of
Tasks. */
/* the current number of tasks in the
list. */
/* the current maximum size of the
dynamic array. */
} TodoList;

/*inserts a new task to the list with
description desc and ONGOING status.
if the list is full, the list should
double in size and insert the new task
returns 1 if successful or 0 if not.*/
int insert(TodoList *todos, const char
*desc);

/*deletes the task at index if
possible. returns 1 if successful or 0
if not.*/
int delete(TodoList *todos, int index);

/*changes the status of task at index
to COMPLETED if possible. returns 1 if
successful or 0 if not.*/
int complete(TodoList *todos, int
index);

/*prints the description and status of
all tasks in the list.*/
void print_list(const TodoList *todos);


/*writes all the tasks of the list into
file fn if possible and returns 1 if
successful or 0 if not.*/
int write_to_file(const TodoList
*todos, const char *fn);


/*frees every allocated space in the
list.*/
void destroy_list(TodoList *todos);
```

todo.c

```
/*include statements*/

/*implement the functions specified in
the header file*/
```

makefile

```
#write all the necessary rules to compile
the previous files
```

Marks

Question	Marks
todo.h	3
todo.c insert	3
todo.c delete	2
todo.c complete	1
todo.c print list	1
todo.c write to file	2
todo.c destroy list	1
makefile	2
Total	15

Sample main

```c
#include <stdio.h>
#include <stdlib.h>
#include "todo.h"

#define INITIAL_SIZE 3

int main(void)
{
    TodoList todos = {NULL, 0, INITIAL_SIZE};

    todos.tasks = malloc(INITIAL_SIZE * sizeof(*todos.tasks));
    if (!todos.tasks) {
        puts("Exit: Cannot allocate initial tasks space.");
        return 1;
    }

    insert(&todos, "Check my Uni email.");
    insert(&todos, "Submit CSC215 lab.");
    insert(&todos, "Buy water from the supermarket.");
    insert(&todos, "Call my grandmother.");
    print_list(&todos);

    complete(&todos, 1);
    print_list(&todos);

    delete(&todos, 2);
    print_list(&todos);

    write_to_file(&todos, "output.txt");

    destroy_list(&todos);

    return 0;
}
```

Sample run

```
>make          (mingw32-make for windows)
gcc -Wall -ansi -c todo.c
gcc -Wall -ansi -c test.c
gcc -Wall -ansi -o myprog todo.o test.o

>./myprog
1- [ONGOING] Check my Uni email.
2- [ONGOING] Submit CSC215 lab.
3- [ONGOING] Buy water from the supermarket.
4- [ONGOING] Call my grandmother.

1- [ONGOING] Check my Uni email.
2- [COMPLETED] Submit CSC215 lab.
3- [ONGOING] Buy water from the supermarket.
4- [ONGOING] Call my grandmother.

1- [ONGOING] Check my Uni email.
2- [COMPLETED] Submit CSC215 lab.
3- [ONGOING] Call my grandmother.

>cat output.txt
1- [ONGOING] Check my Uni email.
2- [COMPLETED] Submit CSC215 lab.
3- [ONGOING] Call my grandmother.
```