UNIVERSITY of GUELPH

CIS2520 Data Structures
Fall 2015, Assignment 1

PART A

Consider three real numbers a, b and c:

if $a \le b$ then $b \ge a$;	(0a)
if $a < b$ then $b > a$;	(0b)
if $a < b$ then $a \le b$;	(0c)
if $a \le b$ and $b \le c$ then $a \le c$;	(0d)
if $a < b$ and $b \le c$ then $a < c$;	(0e)
if $a \le b$ and $b < c$ then $a < c$;	(0f)
if $a < b$ and $b < c$ then $a < c$.	(0g)

When answering the questions below, you may use these basic properties (0a)-(0g) without mentioning them. However, write "according to (1a)" each time you are using (1a), "according to (1b)" each time you are using (1b), etc.

$\forall (a,b,c) \in \mathbb{R}^3,\ (a \le b \rightarrow a+c \le b+c)$	(1a)
$\forall (a,b,c) \in \mathbb{R}^3,\ (a < b \rightarrow a+c < b+c)$	(1b)
$\forall (a,b,c) \in \mathbb{R}^3,\ ((a \le b \wedge c \ge 0) \rightarrow ac \le bc)$	(2a)
$\forall (a,b,c) \in \mathbb{R}^3,\ ((a < b \wedge c > 0) \rightarrow ac < bc)$	(2b)
$\forall (a,b,c,d) \in \mathbb{R}^4,\ ((a \le b \wedge c \le d) \rightarrow a+c \le b+d)$	(3a)
$\forall (a,b,c,d) \in \mathbb{R}^4,\ ((a < b \wedge c \le d) \rightarrow a+c < b+d)$	(3b)
$\forall (a,b,c,d) \in \mathbb{R}^4,\ ((a \le b \wedge c \le d \wedge b \ge 0 \wedge c \ge 0) \rightarrow ac \le bd)$	(4a)
$\forall (a,b,c,d) \in \mathbb{R}^4,\ ((a < b \wedge c \le d \wedge b \ge 0 \wedge c > 0) \rightarrow ac < bd)$	(4b)
$\forall a \in \mathbb{R},\ \cos(a) \ge -1$	(5a)
$\forall a \in \mathbb{R},\ \cos(a) \le 1$	(5b)
$\forall (a,b) \in \mathbb{R}^2,\ ((a \le b \wedge a \ge 0) \rightarrow \sqrt{a} \le \sqrt{b})$	(6a)
$\forall a \in \mathbb{R},\ (a \ge 1 \rightarrow \sqrt{a} \le a)$	(6b)

Note that (1a) is equivalent to $\forall (a,b,c) \in \mathbb{R}^3,\ (b \ge a \rightarrow b+c \ge a+c)$,
(1b) is equivalent to $\forall (a,b,c) \in \mathbb{R}^3,\ (b > a \rightarrow b+c > a+c)$, etc.

QUESTION A.1

0. The following statements are true. Do you understand why?
(You are not asked to answer this question in writing.)

(a) $\exists m \in \mathbb{Z}_+,\ \forall n \in m..+\infty,\ 1 > 0$
(b) $\exists m \in \mathbb{Z}_+,\ \forall n \in m..+\infty,\ \sqrt{n} > 0$
(c) $\exists m \in \mathbb{Z}_+,\ \forall n \in m..+\infty,\ n > 0$
(d) $\exists m \in \mathbb{Z}_+,\ \forall n \in m..+\infty,\ n^2 > 0$

1. Show that:

(a) $\exists m \in \mathbb{Z}_+,\ \forall n \in m..+\infty,\ 5-100/n > 0$
(b) $\exists m \in \mathbb{Z}_+,\ \forall n \in m..+\infty,\ 3\sqrt{n}+\cos(n) > 0$
(c) $\exists m \in \mathbb{Z}_+,\ \forall n \in m..+\infty,\ 2n-100 > 0$
(d) $\exists m \in \mathbb{Z}_+,\ \forall n \in m..+\infty,\ 5n^2-8n+2 > 0$

2. Show that:

(a) $\exists \lambda \in \mathbb{R}_+,\ \exists m \in \mathbb{Z}_+,\ \forall n \in m..+\infty,\ 5-100/n \leq \lambda$
(b) $\exists \lambda \in \mathbb{R}_+,\ \exists m \in \mathbb{Z}_+,\ \forall n \in m..+\infty,\ 3\sqrt{n}+\cos(n) \leq \lambda\sqrt{n}$
(c) $\exists \lambda \in \mathbb{R}_+,\ \exists m \in \mathbb{Z}_+,\ \forall n \in m..+\infty,\ 2n-100 \leq \lambda n$
(d) $\exists \lambda \in \mathbb{R}_+,\ \exists m \in \mathbb{Z}_+,\ \forall n \in m..+\infty,\ 5n^2-8n+2 \leq \lambda n^2$

QUESTION A.2

Consider four functions f_1, f_2, g_1 and g_2 from $\mathbb{Z}_+$ to $\mathbb{R}_+$. Assume there exist two positive real numbers λ_1 and λ_2 and two positive integers m_1 and m_2 such that:

$\forall n \in m_1..+\infty,\ f_1(n) \leq \lambda_1 g_1(n)$
$\forall n \in m_2..+\infty,\ f_2(n) \leq \lambda_2 g_2(n)$

Show that: $\exists \lambda \in \mathbb{R}_+,\ \exists m \in \mathbb{Z}_+,\ \forall n \in m..+\infty,\ f_1(n)+f_2(n) \leq \lambda(g_1(n)+g_2(n))$

HARD COPY SUBMISSION

Check the course outline for instructions.

MARKING SCHEME

A.1 = 24% and A.2 = 6%

PART B

QUESTION B.1: *Sequential implementation of a list*

Store the nine text files in a folder **List_Student_S**. Read all the **.h** files, and then complete the files **StudentImplementation.c** and **ListImplementation.c** where indicated. Include instructions to test the operations' preconditions and postconditions. These instructions should be compiled only when in debug mode, using the `#define`, `#ifdef` and `#endif` preprocessor directives.

QUESTION B.2: *Test program*

Complete the file **myProgram.c** where indicated. This simple test program uses the functions declared in **StudentInterface.h** and **ListInterface.h** to create and modify a list of students as specified in **test.txt**. With the current **test.txt** file, the output of the program should be the following:

```
List is empty; list is not full; list is of size 0:
List is not empty; list is not full; list is of size 1:
     John  75%
List is not empty; list is not full; list is of size 2:
     Mary  80%
     John  75%
List is not empty; list is not full; list is of size 3:
     Mary  80%
     John  75%
     Pete  90%
List is not empty; list is full; list is of size 4:
     Mary  80%
     John  75%
     Liz   85%
     Pete  90%
List is not empty; list is not full; list is of size 3:
     John  75%
     Liz   85%
     Pete  90%
List is not empty; list is not full; list is of size 2:
     John  75%
     Pete  90%
List is not empty; list is not full; list is of size 1:
     John  75%
List is empty; list is not full; list is of size 0:
```

QUESTION B.3: *Linked implementation of a list*

Make a copy **List_Student_L** of the folder **List_Student_S**. In that new folder, modify **StudentType.h** as shown below; modify **StudentImplementation.c** accordingly; modify **ListType.h** and **ListImplementation.c** to accommodate a one-way linked list representation instead of a sequential list representation. The other files should not be modified, and the output of your test program should be exactly the same except for `list is full`, which should not happen.

StudentType.h

```
typedef struct {
      char *name;
      int grade;
} Student;
```

ELECTRONIC SUBMISSION

Create a root folder **CIS2520_LastNameFirstName_A1** with a **README.txt** text file and two subfolders: **List_Student_S** (with **test.txt**, the **makefile** and all the source and header files for the sequential implementation) and **List_Student_L** (with **test.txt**, the **makefile** and all the source and header files for the linked implementation). All the files in these folders must be text files. Do not submit any object or executable files. The two sets of source files must be compilable independently with *GCC 4.4.0* or later versions, using the provided **makefile**. Zip the root folder **CIS2520_LastNameFirstName_A1** and upload it to *Moodle*. Check the course outline for additional instructions.

MARKING SCHEME

Sequential implementation, not including preconditions and postconditions (B.1) = 25%
Linked implementation, not including preconditions and postconditions (B.3) = 25%
Preconditions and postconditions (B.1 and B.3) = 10%
Test program (B.2) = 10%