CIS2520 Data Structures

Fall 2015, Assignment 2

PART A

Consider the properties (1)-(11) below. When answering the questions in A1 and A2, write “according to (1)” each time you are using (1), “according to (2)” each time you are using (2), etc.

$\forall(a,b,c)\in\mathbb{R}^3$, $(a\leq b \rightarrow a+c\leq b+c)$ (1)

$\forall(a,b,c)\in\mathbb{R}^3$, $((a\leq b\wedge c\geq 0) \rightarrow ac\leq bc)$ (2)

$\forall(a,b,c,d)\in\mathbb{R}^4$, $((a\leq b\wedge c\leq d) \rightarrow a+c\leq b+d)$ (3)

$\forall(a,b,c,d)\in\mathbb{R}^4$, $((a\leq b\wedge c\leq d\wedge b\geq 0\wedge c\geq 0) \rightarrow ac\leq bd)$ (4)

$\forall n\in\mathbb{Z}_+$, $\exists k\in\mathbb{Z}_+$, $2^{k-1}\leq n\leq 2^k$ (5)

$\forall n\in\mathbb{Z}_+$, $\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$ (6)

The natural logarithm, log, is a total function from $\mathbb{R}_+$ to $\mathbb{R}$ such that:

$\log(1)=0$ (7)

$\forall(a,b)\in\mathbb{R}_+^2$, $\log(ab) = \log(a)+\log(b)$ (8)

$\forall a\in\mathbb{R}_+$, $\forall k\in\mathbb{N}$, $\log(a^k) = k\log(a)$ (9)

log is increasing on $\mathbb{R}_+$ (10)

The binary logarithm, $\log_2$, is the total function from $\mathbb{R}_+$ to $\mathbb{R}$ defined by:

$\forall a\in\mathbb{R}_+$, $\log_2(a) = \log(a)/\log(2)$ (11)

A1

Show that the algorithm below runs in $O(n^2)$ time, where n denotes the input size.

```
function someFunction (A)
    for j=1 to A.length-1
        for i=1 to A.length-j
            if A[i-1] > A[i]
                swap A[i-1] and A[i]
```

A2

Let T be a total function from $\mathbb{Z}_+$ to $\mathbb{R}_+$ such that T is nondecreasing on $\mathbb{Z}_+$,
$T(1)=7$, and: $\forall k \in \mathbb{Z}_+,\ T(2^k)=T(2^{k-1})+5$

a) Calculate T(2), T(4) and T(8).
b) Prove by induction that: $\forall k \in \mathbb{N},\ T(2^k)=5k+7$
c) Let k and n be two positive integers such that $2^{k-1} \leq n \leq 2^k$.
Show that $2^k \leq 2n$ and that $k \leq \log_2(2n)$.
d) Show that for any positive integer n we have $T(n) \leq 5\log_2(2n)+7$.
e) Show that T(n) is $O(\log(n))$.

HARD COPY SUBMISSION

Check the course outline for instructions.

MARKING SCHEME

A1 = 20%
A2 = 20%

PART B

B1

Download **A1key.zip**. *It packs two folders:* **List_Student_S** *and* **List_Student_L**.
This question concerns the files in the folder **List_Student_L**.

a) In **myProgram.c**, replace

```
#include "ListInterface.h"
```

with

```
#include "StudentInterface.h"
#include "ListInterface.h"
```

Including both header files causes a compilation error, because **ListInterface.h** already includes **StudentInterface.h** (through **ListType.h**), and the type **Student** is therefore defined twice (which is not allowed in C). A way to go around this is to use *#include guards*: add appropriate **#ifndef**, **#define** and **#endif** directives in **StudentInterface.h** so that **myProgram.c** compiles.

b) Do you know the **assert()** macro? Well, if you don't, you can easily find out on your own, can't you? Modify **StudentImplementation.c** and **ListImplementation.c** so that the pre- and post- conditions are checked when in debug mode. Do not use the **#define** directive, however, and do not use the **printf()** and **exit()** functions. Use the **assert()** macro instead, and modify the **makefile** so that

```
make –B
```

unconditionally makes all targets with debugging ON
(the pre- and post- conditions are checked), while

```
make –B FLAG=–DNDEBUG
```

unconditionally makes all targets with debugging OFF
(the pre- and post- conditions are not checked).

B2

Make a copy **Stack_int_L** *of the revised folder* **List_Student_L**.
This question concerns the files in **Stack_int_L**.

a) Delete the files **StudentType.h**, **StudentInterface.h** and **StudentImplementation.c**.

b) In **ListType.h**, replace

```
#include "StudentInterface.h"
typedef Student Item;
```

with

```
typedef int Item;
```

c) In **ListInterface.h** , replace

```
#include "ListType.h"
```

with

```
#include "StackType.h"
```

and replace the function declarations with

```
extern void Initialize (Stack *S);
extern void Push (Item X, Stack *S);
extern void Pop (Stack *S);
extern int Full (Stack *S);
extern int Empty (Stack *S);
extern int Size (Stack *S);
extern void Top (Stack *S, Item *X);
extern void Destroy (Stack *S);
```

d) Rename **ListType.h**, **ListInterface.h** and **ListImplementation.c**:
call them **StackType.h**, **StackInterface.h** and **StackImplementation.c**.

e) Replace **test.txt** with:

test.txt

```
63761203947939841001998398359383983929210126738495 01
45228016205639283740909281372304758 60
```

f) Modify all the files according to the changes above, and so that the program (**a.out**) displays the sum of the two numbers that are stored in **test.txt.** This sum should be calculated using three stacks, as shown in class.

ELECTRONIC SUBMISSION

Make sure the revised folders **List_Student_L** and **Stack_int_L** contain text files only (**.h**, **.c**, **makefile**, **test.txt**). Make sure all the file and function header comments have been updated according to the requested changes. Place the two folders along with a **README.txt** text file in a root folder **CIS2520_LastNameFirstName_A2**. Zip the root folder and upload it to *Moodle*. Check the course outline for additional instructions.

MARKING SCHEME

B1 = 20%
B2 = 40%