M263 Solutions For 2004 (Issue 1.0)

PART I

Question 1 (4 Marks)

(a) 4
(b) [1,2,3]
(c) **false**

Question 2 (6 Marks)

(a)

I	anInt
1	1
2	6
3	16

The value 16 is returned.

(b) **pre** The array arr is full and SIZE(arr) ≥ 3.

Question 3 (5 Marks)

(a) **function** QU3(arr **in Array of Int**) **return in** *Bool*
pre The array arr is full.
post The returned value is **true** if the 1st and last numbers in the array arr are the same, and **false** otherwise. For the arrays [1,2,1] and [1] **true** is returned. For the array [1,2] **false** is returned.

(b)

```
function QU3(arr)
{
var aBool in Bool
var first in Int
var last in Int
first ← AT(1,arr)
last ← AT(SIZE(arr),arr)
aBool ← (first == last)
return aBool
}
```

OR

```
function QU3(arr)
{
return (AT(1,arr) == AT(SIZE(arr),arr))
}
```

Question 4 (4 Marks)

After (1) the state of s is (4,-2).
After (2) the state of s is (14,-4).

Question 5 (3 Marks)

t.store()
t.penOn()
t.forward(5)
t.restore()

Question 6 (4 Marks)

Comment	**v**	**s**
Before loop	[]	"a"
After (1)	["a"]	"aq"
After(1)	["aq","a"]	"aqq"
After (2)	["a","aq"]	"aqq"

Question 7 (4 Marks)

(a)(i) {11, 14, 15, 21, 22, 23}

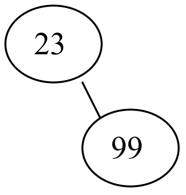
(a)(ii) {14, 21}

(a)(iii) {11, 15}

(b) 6

Question 8 (5 Marks)

(a)



(b) Not valid. root is not a method of BinTree..
(c) Not valid. t.isEmpty() returns a Boolean value but leftTree is not a method of Bool.

Question 9 (4 Marks)

a b	**3 2 1** **b ∧ ¬ (a ∧ b)**	**b ∧ ¬ (a ∧ b)**
T T	T F F T T T	F
T F	F F T T F F	F
F T	T T T F F T	T
F F	F F T F F F	F

The disjunctive normal form is ¬ a ∧ b.

Question 10 (5 Marks)

(a) Sam is a girl younger than 9.

(b) At least one boy in the class is younger than 9.

(c) Sam is not taller than any of the boys in the class.

Question 11 (3 Marks)

{custId} ~→ {custAdd}

{contact} ~→ {staffExt}

{custId} ~→ {contact}

Question 12 (5 Marks)

	Statement used
QU12([-10, -5, 8, 2])	
= 1 + QU12([-5, 8, 2])	3
= 1 + (1 + QU12([8, 2])	3
= 1 + (1 + 1)	2
=3	

Note. If the vector contains a positive integer then returns position of 1st positive integer, otherwise returns (size of vector + 1).

Question 13 (5 Marks)

(a) Just before the loop is entered then

count has value 0 from the assignment count ← 0

toDo has the value s from the assignment toDo ← s

Therefore count + COUNTPOS(toDo) has the value 0 + COUNTPOS(s). As this equals COUNTPOS(s) then the LIC holds just before the while loop is entered.

(b) The while loop will stop executing when size(toDo) == 0. As the stack is empty then COUNTPOS(toDo) is 0. Therefore using the LIC we have count = COUNTPOS(s).

Therefore the value returned by impCountPos is correct.

Question 14 (3 Marks)

(a) f is $\Theta(n \log_2(n))$

g is $\Theta(n^2)$

(b) For large n implementation A is more efficient as $\Theta(n \log_2(n)) \subset \Theta(n^2)$.

PART II

Question 15 (20 Marks)

(a) 7 marks

(a)(i) TENNIS(x) ⇒ ¬ MATHS(x)

≡ ¬ TENNIS(x) ∨ ¬ MATHS(x)	Rewriting ⇒ (i)
≡ ¬ (TENNIS(x) ∧ MATHS(x))	DeMorgan (ii)

(a)(ii) ¬ (∀s in Students: [TENNIS(x) ⇒ ¬ MATHS(x)])

Some students in the school tennis team have passed maths A level.
An equivalent formal proposition is ∃s in Students: [TENNIS(x) ∧ MATHS(x)]

¬ (∀s in Students: [TENNIS(x) ⇒ ¬ MATHS(x)])

≡ ∃s in Students: [¬ (TENNIS(x) ⇒ ¬ MATHS(x))]	Extended DeMorgan (i)
≡ ∃s in Students: [¬¬ (TENNIS(x) ∧ MATHS(x))]	Using part (i)
≡ ∃s in Students: [TENNIS(x) ∧ MATHS(x)]	Negation (iii)

(b) 9 marks

Let $aSpot_j$ represent the state of aSpot after j executions of the for loop.
Let p(j) be the proposition that $aSpot_j$ is (2j, j), where the integer $j \geq 0$.

Initial condition. The default state of a spot is (0, 0). Therefore before the loop is executed (j = 0) the state of the spot is (0, 0). Therefore p(0) is true.

Inductive step. For j > 0, assume that after j - 1 executions of the loop that p(j - 1) is true. Therefore after j - 1 executions of the loop the state of $aSpot_{j-1}$ is (2(j - 1), j - 1). The effect of the 3 statements in the loop is to change the state of aSpot to (2j - 2, j), (2j - 1, j), (2j, j) respectively. Therefore $aSpot_j$ is (2j, j) and so p(j) is true.

Therefore by the principle of mathematical induction p(j) is true for all integers $j \geq 0$.

Since the loop in the fragment is executed n times then the state of aSpot is (2n, n).

(c) 4 marks

jack ∧ jill	(1)	premise
jack ⇒ fall	(2)	premise
¬fall ∨ spilt	(3)	premise
jack	(4)	simplification 1
fall	(5)	implication 2,4
¬¬fall	(6)	Equivalent to (5) using negation (iii)
spilt	(7)	disjunction 3,6

Hence $\dfrac{\text{jack} \wedge \text{jill}; \text{jack} \Rightarrow \text{fall}; \neg\text{fall} \vee \text{spilt}}{\text{spilt}}$.

Question 16 (20 Marks)

(a) 7 marks

(1) found ← false

(2) i ← 1 to s.length()

(3) s.charAt(i) == c

(4) found ← true

(b) (4marks)

∃i in Int [i > 0 ∧ i ≤ s.length() ∧ s.charAt(i) == c]

(c) (5 marks)
(1) s.length() == 0
(2) b ← false
(3) b ← ((s.charAt(1) == c) ∨ ISIN(s.substr(2, s.length() - 1), c))

(d) (4 marks)

```
QU16 extends mString

method charIn(c)
{
var i in Int
for(i ← 1 to this.length())
{
 if (this.charAt(i) == c)
 { return true}
}
return false
}
```

Question 17 (20 Marks)

(a) (3 marks)

(a)(i) t_1, t_2.
(a)(ii) t_1, t_5.
(a)(iii) t_3, t_4.

(b) (2 marks)

Rule	Resulting Dependency
1	FD1. {courseID} ~→ {date}
2	FD2. {studID, courseID} ~→ {venue}
3	FD3. {venue} ~→ {contactID}
4	FD4. {contactID} ~→ {contactPhone}
5	FD5. {contactPhone} ~→ {contactID}

(c) (3 marks)

Let K = {studID, courseID}.

{courseID} ⊂ K determines date (FD1). *Correction by Judith Essex.*
K determines venue.
K transitively determines contactID (FD2, FD3).
K transitively determines contactPhone (FD2, FD3, FD4).

Therefore K had the determining property. Parts (a)(i) and (a)(ii) show that K has the minimal property. Therefore {studID, courseID} is a key of EXAM.

(d) (3 marks)

{studID, courseID} ~→ {venue} and {venue} ~→ {contactID}.

(e) (3 marks)

{courseId} is a proper subset of the key and {courseID} ~→ {date}. Since a proper subset of the key determines an attribute not in the key then EXAM is not in second normal form.

We can divide EXAM into the following 2 schemes.

COURSE
Attributes : courseID, date.
Functional Dependencies : FD1. Key : courseID

EXAM_A
Attributes : studID, courseID, venue, contactID, contactPhone.
Functional Dependencies : FD2, FD3, FD4, and FD5. Key : studID, courseID

COURSE is in second normal form as it only contains 2 attributes.
EXAM_A is in second normal form as it has key {studID, courseID} and no subset of the key determines an attribute not in the key.

(f) (3 marks)

COURSE is in third normal form as it only contains 2 attributes.

In the scheme EXAM_A there are the following transitive dependencies

{studID, courseID} ~→ {venue}, {venue} ~→ {contactID};
{venue} ~→ {contactID}, {contactID} ~→ {contactPhone}.

We can divide EXAM_A into the following 2 schemes.

CONTACT
Attributes : contactID, contactPhone.
Functional Dependencies : FD14, FD5. Key : contactID

EXAM_B
Attributes : studID, courseID, venue, contactID.
Functional Dependencies : FD2, FD3. Key : studID, courseID

In the scheme EXAM_B there is the following transitive dependency

{studID, courseID} ~→ {venue}, {venue} ~→ {contactID};

We can divide EXAM_B into the following 2 schemes.

VENUE
Attributes : venue, contactID.
Functional Dependencies : FD3. Key : venue

EXAM_C
Attributes : studID, courseID, venue.
Functional Dependencies : FD2. Key : studID, courseID

(g) (2 marks)

One to many association between EXAM_C and COURSE with foreign key {courseID}.
One to many association between EXAM_C and VENUE with foreign key {venue}.
One to one association between VENUE and CONTACT with foreign key {contactID}.

Thanks to Joe McCabe for help with this question.

Question 18 (20 Marks)

(a) 4 marks

Statement used	**Comment**
1	3 == 0 is false
3	5 == 10 is false
5	5 > 10 is false
7	Call TENINBST with tree (10 (9 ()()) (15 ()()))
1	2 == 0 is false
3	10 == 10 is true
4	Returns true

Number of statement executions is 7.

(b) (10 marks)

(i) The tree does not contain the value 10 and a node at each of the n levels of the BST has to be checked to determine this.

(ii) If the depth of the tree is 0 then only statements 1 and 2 will be executed.
Therefore T(0) = 2.

To process a tree of depth n (n > 0) the value at the root node is checked and if this is not 10 then a BST of depth n-1 has to be checked. We execute statements 1, 3, 5, and then either 6 or 7 (4 statements). Statements 6 and 7 both cause a tree of depth n-1 to be processed which requires a further T(n - 1) statements to be executed.

Therefore T(n) = T(n-1) + 4.

(iii)
$$
\begin{aligned}
T(n) &= T(n-1) + 4\\
T(n-1) &= T(n-2) + 4\\
&\ldots\ldots\ldots\ldots\\
T(2) &= T(1) + 4\\
T(1) &= T(0) + 4\\
T(0) &= 2
\end{aligned}
$$

Adding the above equations and cancelling the terms which occur on both sides of the resulting equation gives T(n) = 4n + 2.

Therefore T(n) is $\Theta(n)$.

(iv) T(3) = 4 * 3 + 2 = 14.
As T(3) is for the worst case then it must be greater than or equal to the value found in part (a). Since the tree in part (a) contains the value 10 then it is not a worse case tree so we would expect this value to be less than 14. This is the case.

(c) (6 marks)

Initial condition. F(0) = 5 * 2^0 - 3 = 2. Therefore U0) = F(0).

Inductive step.

Assume for k > 0 that U(k - 1) = F(k - 1) = 5 * 2^{k-1} - 3.

U(k) = 2 * U(k - 1) + 3
= 2 * (5 * 2^{k-1} - 3) +3
= 5 * 2^k - 3 = F(k)

This establishes the inductive step.

Therefore by the principle of mathematical induction U(n) = F(n) for all integers n ≥ 0.