PLEASE HAND IN

PLEASE HAND IN

UNIVERSITY OF TORONTO
Faculty of Arts and Science

DECEMBER 2017 EXAMINATIONS

CSC 343 H1F
Instructor: Craig, Horton & Meraji

Duration — 3 hours

Aids Allowed: None

Student Number: |_|_|_|_|_|_|_|_|_|_|

Family Name(s): ____________________

Given Name(s): ____________________

# 1:	______/12
# 2:	______/ 7
# 3:	______/15
# 4:	______/ 4
# 5:	______/ 4
# 6:	______/ 6
# 7:	______/ 4
# 8:	______/ 6
# 9:	______/ 5
# 10:	______/ 5
# 11:	______/ 3
# 12:	______/ 4
# 13:	______/ 8
# 14:	______/ 7
# 15:	______/ 5
TOTAL:	______/95

In the SQL questions, you are welcome to use views. Comments are not required, although they may help us mark your answers.

There are three blank pages at the end for rough work.

A mark of at least 40 out of 100 on this exam is required in order to pass the course.

It's been a real pleasure teaching you this term. Good luck!

Question 1. [12 MARKS]

Consider this schema for an information system about hospital admissions at a single hospital. Keys are underlined and null values are not allowed. All "time" attributes include date and time.

Relations

Patients(id, DOB)
DOB is the patient's date of birth.

Admissions(admitID, pid, admitTime, dischargeTime)
This patient was admitted to hospital at *admitTime* and discharged at *dischargeTime.*

ICUStays(icuStayID, admitID, inTime, outTime)
During admission *admitID,* the patient was in the Intensive Care Unit from *inTime* to *outTime.*

Diagnoses(id, pid, admitID, code)
During admission *admitID,* patient *pid* received the diagnosis with numeric code *code.*

Integrity constraints

Admissions[pid] $\subseteq$ Patients[id]

ICUStays[admitID] $\subseteq$ Admissions[admitID]

Diagnoses[pid] $\subseteq$ Patients[id]

Diagnoses[admitID] $\subseteq$ Admissions[admitID]

Part (a) [1 MARK]

How many times may a patient stay in the ICU during a single hospital admission? Circle all possible answers.

0 1 2 5

Part (b) [1 MARK]

Assuming that there are no other functional dependencies other than those implied by the keys, do the FDs of every relation satisfy the BCNF property? Circle your answer and provide an explanation.

YES NO

Explanation:

Part (c) [3 MARKS]

The schema includes only a numeric code for each diagnosis. Suppose that for every diagnosis in the Diagnoses table, we want to record both a short name (such as 'Staph food poisoning') and a long name (such as 'Staphylococcal food poisoning'). Describe how you would change the schema to accomplish this.

Part (d) [2 MARKS]

Suppose we add the following integrity constraint to the schema.

$$\sigma_{(inTime<admitTime)\vee(outTime>dischargeTime)}[ICUStays \bowtie Admissions] = \emptyset$$

Explain what this constraint ensures.

Provide example data that violates this constraint. (We have filled in one value for you.)

Admissions:

admitID	pid	admitTime	dischargeTime
		Jul 1, 2016 @9pm	

ICUStays:

icuStayID	admitID	inTime	outTime

Part (e) [5 MARKS]

Some hospital admissions involve ICU stays but others (termed non-critical) do not involve a stay in the ICU. Write a relational algebra query to find pids of patients who have had more than two non-critical admissions to the hospital.

Question 2. [7 MARKS]

This question continues with the same schema. Here it is again for your convenience.

Relations

Patients(id, DOB)
DOB is the patient's date of birth.

Admissions(admitID, pid, admitTime, dischargeTime)
This patient was admitted to hospital at *admitTime* and discharged at *dischargeTime.*

ICUStays(icuStayID, admitID, inTime, outTime)
During admission *admitID*, the patient was in the Intensive Care Unit from *inTime* to *outTime.*

Diagnoses(id, pid, admitID, code)
During admission *admitID*, patient *pid* received the diagnosis with numeric code *code.*

Integrity constraints

Admissions[pid] $\subseteq$ Patients[id]

ICUStays[admitID] $\subseteq$ Admissions[admitID]

Diagnoses[pid] $\subseteq$ Patients[id]

Diagnoses[admitID] $\subseteq$ Admissions[admitID]

Part (a) [3 MARKS]

Neonatal admissions are admissions of a patient who is less than 1 year old at the time of admission. We want to know the diagnosis for each patient's most recent neonatal admission. Below is a relational algebra query that is syntactically correct, but it does not give the correct answer in all cases.

Assume that the selection criterion "$(admitTime - DOB) < 1year$", used below, correctly selects patients who were less than 1 year old at the time of this hospital admission.

$$Neonates(pid) := \pi_{id}\sigma_{Patients.id=Admissions.pid \wedge (admitTime-DOB)<1year}[Patients \times Admissions]$$

$$NeoDiagnosis(pid, code, admitTime) := \pi_{pid,\ code,\ admitTime}[Neonates \bowtie Admissions \bowtie Diagnoses]$$

$$NotRecent(pid, code, admitTime) :=$$

$$\pi_{N2.pid, N2.code, N2.admitTime}\sigma_{\substack{N2.admitTime<N1.admitTime \\ \wedge \\ N1.pid=N2.pid}}[\rho_{N1}NeoDiagnosis \times \rho_{N2}NeoDiagnosis]$$

$$MostRecent(pid, code, admitTime) := NeoDiagnosis - NotRecent$$

Explain what is wrong with the query and, on the next page, give a specific example of where it fails.

Explanation:

Data on which the query fails (we filled in one value for you):

Patients:

id	DOB

Admissions:

admitID	pid	admitTime	dischargeTime
		Jul 1, 2016 @9pm	

Diagnoses:

id	pid	admitID	code

Incorrect result obtained by the query:

pID	code	admitTime

Part (b) [1 MARK]

We are interested in the percentage of time during an admission that a patient spends in the ICU. We will call this the **per-admission ICU percentage**. With our current schema, we are unable to calculate this value using relational algebra. Briefly explain why.

Part (c) [3 MARKS]

If we were using SQL rather than relational algebra could we calculate the per-admission ICU_Percentage? Circle one answer and either explain briefly in English why it can't be done, or give a brief overview in English of how it could be solved in SQL.

YES NO

Explanation/Overview:

Question 3. [15 MARKS]

Recall the schema below, which we have used in the course this term.

Relations

Department (dID, name, division)

Employee(eID, name, salary, dept)

Manages(manager, junior)
Employee *junior* reports directly to *manager*

Sales(eID, day, amount)
Employee *eID* had sales totalling *amount* on *day*

Integrity constraints

Employee[dept] $\subseteq$ Department[dID]

Manages[manager] $\subseteq$ Employee[eID]

Manages[junior] $\subseteq$ Employee[eID]

Sales[eID] $\subseteq$ Employee[eID]

Part (a) [4 MARKS]

Write a query in SQL that finds managers who directly manage more than four employees. Report the manager name and the average sales in the year 2009 across all the people they manage.
Reminder: Use `EXTRACT (year FROM <timestamp element>)` to get the year component of a date.

Part (b) [2 MARKS]

Management is interested in closing small departments and merging medium ones. First create a view named `DepartmentSize` that has department name, department ID, and the number of employees in that department. Hint: Don't forget to include departments that exist but have zero employees.

Part (c) [3 MARKS]

Using your `DepartmentSize` view, now write an SQL query to find departments with at most 6 employees. Report the department name, department id, num_employees and size, where size is either 'small' (fewer than 3 employees) or 'medium' (3 to 6, inclusive).

Part (d) [6 MARKS]

Write an SQL query that reports the eID of employees whose monthly sales (for any months in which they had any sales at all) never decreased in 2017. Include employees who have had only *one* month with sales in 2017 but do not include those who had no sales in 2017.

Question 4. [4 MARKS]

We continue with the same schema from the previous question.

Relations

Department (dID, name, division)

Employee(eID, name, salary, dept)

Manages(manager, junior)
Employee *junior* reports directly to *manager*

Sales(eID, day, amount)
Employee *eID* had sales totalling *amount* on *day*

Integrity constraints

Employee[dept] ⊆ Department[dID]

Manages[manager] ⊆ Employee[eID]

Manages[junior] ⊆ Employee[eID]

Sales[eID] ⊆ Employee[eID]

Consider the following two queries:

```
-- Query 1:
SELECT manager
FROM manages M1
WHERE exists (
   SELECT *
   FROM Manages M2 JOIN Sales ON junior = eId
   WHERE M1.manager = M2.manager AND
         amount > 90);
```

```
-- Query 2:
SELECT manager
FROM manages M1
WHERE exists (
   SELECT *
   FROM Sales
   WHERE junior = eID AND
         amount > 90);
```

These queries are similar but not equivalent. Create instances of Manager and Sales, with no more than 4 rows each, and for which these queries produce different results.

Manages: manager junior   Sales: eid day amount

Now show the results of the queries on this data.

Result of Query 1:

Result of Query 2:

Question 5. [4 MARKS]

This question continues with the same schema from the previous question.

There are many metrics we'd like to compute from our data. Some may not be computable in every language we have studied. In the table below, circle YES or NO to indicate which quantities can be computed using which languages.

Here are some definitions we will use in this question:

- A manager M's **second-level reportees** are analogous to grandchildren; they are employees whose managers report to M.
- An employee who does not manage anyone else is at management level 0. A manager who manages only employees at level 0, is at level 1. A manager's **management level** is 1 + the maximum management level of their direct employees.
- A **employee's bonus** is 10% of their own annual sales plus 1% of the bonus of every employee they directly supervise as a manager.
- The **department salary ratio** is the ratio of highest salary over lowest salary for employees in that department.
- The **department sales ratio** is the ratio of highest annual sales over lowest annual sales for employees in that department for a particular year.

Quantity of interest	Relational Algebra	SQL	Embedded JDBC
the day in 2010 which had the largest total sales	yes no	yes no	yes no
the second-level reportees of a specific manager	yes no	yes no	yes no
the highest level of management in the organization	yes no	yes no	yes no
the bonus amount for every employee	yes no	yes no	yes no
the lowest salary in the organization	yes no	yes no	yes no
the second highest salary in the organization	yes no	yes no	yes no
department salary ratios for every department	yes no	yes no	yes no
department sales ratios for every department for 2016	yes no	yes no	yes no

Question 6. [6 MARKS]

Consider the following schema:

```
CREATE TABLE One (
   a INT PRIMARY KEY,
   b INT UNIQUE
);

CREATE TABLE Two (
   c INT PRIMARY KEY,
   d INT,
   e INT,
   FOREIGN KEY (c) REFERENCES One(a) ON UPDATE SET NULL ON DELETE CASCADE
);

CREATE TABLE Three (
   f INT,
   g INT PRIMARY KEY,
   FOREIGN KEY (g) REFERENCES One(b) ON UPDATE CASCADE ON DELETE CASCADE,
   FOREIGN KEY (f) REFERENCES Two(c) ON UPDATE RESTRICT ON DELETE CASCADE
);

CREATE TABLE Four (
   h INT,
   i INT,
   j INT
);
```

Part (a) [4 MARKS]

Suppose the tables have been populated as shown below. Modify the data to show the contents of the tables after this command is executed:

```
DELETE from One WHERE b = 5;
```

```
One:               Two:               Three:             Four:
 a | b              c | d | e          f | g              h | i | j
---+---            ---+---+---        ---+---            ---+---+---
 4 | 2              5 | 1 | 2          5 | 1              0 | 3 | 1
 7 | 4              4 | 2 | 5          1 | 2              1 | 1 | 1
 3 | 3              7 | 3 | 9          7 | 3              2 | 1 | 3
 1 | 5              6 | 4 | 8          6 | 4              4 | 4 | 0
 5 | 1              1 | 5 | 7          5 | 5
 6 | 0                                 1 | 0
```

Part (b) [2 MARKS]

Consider what would happen if we started with the tables as originally populated:

```
One:
 a | b
---+---
 4 | 2
 7 | 4
 3 | 3
 1 | 5
 5 | 1
 6 | 0
```

```
Two:
 c | d | e
---+---+---
 5 | 1 | 2
 4 | 2 | 5
 7 | 3 | 9
 6 | 4 | 8
 1 | 5 | 7
```

```
Three:
 f | g
---+---
 5 | 1
 1 | 2
 7 | 3
 6 | 4
 5 | 5
 1 | 0
```

```
Four:
 h | i | j
---+---+---
 0 | 3 | 1
 1 | 1 | 1
 2 | 1 | 3
 4 | 4 | 0
```

and executed this syntactically correct command:

```
UPDATE One SET b = -1 WHERE a = 7;
```

Is it possible to add a reaction policy to Table Four, and change nothing else, so that afterwards Table Four would contain this:

```
 h | i | j
---+---+---
 0 | 3 | 1
 1 | 1 | 1
 2 | 1 | 3
 4 |   | 0
```

Circle one: Yes No

If yes, add to the definition of the tables on the previous page any additional pieces of reaction policy that must have been in place. If no, explain briefly why not (be brief; you will not need this much space).

Question 7. [4 MARKS]

Consider relation $R(A, B, C, D, E)$ with FDs

$$S = \{\ D \rightarrow AC,\ \ A \rightarrow B,\ \ C \rightarrow ED,\ \ E \rightarrow AB\ \}.$$

Suppose you want to decompose R into BCNF. For each FD, indicate whether it violates BCNF. If it does, give the attributes that would be in R1 and R2 if you decomposed R into R1 and R2 based that FD.

Dependency	Violates BCNF?	R1 attributes	R2 attributes
$D \rightarrow AC$	YES NO		
$A \rightarrow B$	YES NO		
$C \rightarrow ED$	YES NO		
$E \rightarrow AB$	YES NO		

Question 8. [6 MARKS]

Consider relation $R(J, K, L, M, N)$ with FDs

$$S = \{\ N \rightarrow M,\ \ JM \rightarrow L,\ \ L \rightarrow M\ \}.$$

Suppose we plan to decompose R into three relations as follows: $R1(J, L, N)$, $R2(J, K, N)$, $R3(J, K, M)$.

Part (a) [2 MARKS]

Calculate the projection of S onto $R1$. Show your work.

Part (b) [4 MARKS]

Now you will use the chase test to determine if this decomposition is lossless or lossy. First show the initial tableau based on the decomposition, before applying the FDs.

Next, copy the tableau out again below (so we can mark the untouched one above) and complete the Chase Test on this second copy.

What is your final conclusion? This decomposition is (circle one): LOSSLESS LOSSY.

If you concluded it was lossy, give an instance of R that demonstrates this, that is, one for which $R \neq R1 \bowtie R2 \bowtie R3$.

Question 9. [5 MARKS]

Consider relation $R(A, B, C, D, E, F)$ with functional dependencies:

$$S = \{\ E \rightarrow DB,\ \ BC \rightarrow A,\ \ F \rightarrow E,\ \ D \rightarrow C\ \}$$

In this question you will perform 3NF synthesis on R.

Part (a) [1 MARK]

Give a minimal basis for S. Write your rough work (which will not be marked) below, and put the minimal basis in the box on the next page. If S already comprises a minimal basis, just state this.

A minimal basis is:

Part (b) [2 MARKS]

List the relations generated based only on the FDs. (We will do the final step of the 3NF synthesis algorithm in the next question.)

Part (c) [2 MARKS]

Do we need to add an extra relation? Circle one: YES NO

If you answered yes, list the attributes of one acceptable extra relation that we could add to finish the 3NF synthesis. If you answered no, justify your answer.

Question 10. [5 MARKS]

Consider the relation $R(J, K, L, M, N)$ with the following functional dependencies:

$$\{\, KL \rightarrow N,\;\; J \rightarrow M,\;\; N \rightarrow M,\;\; LM \rightarrow K \,\}.$$

Part (a) [1 MARK] What is the closure of KL?

Part (b) [1 MARK] Do K and L together form a superkey for R? Circle one answer and explain.

YES NO

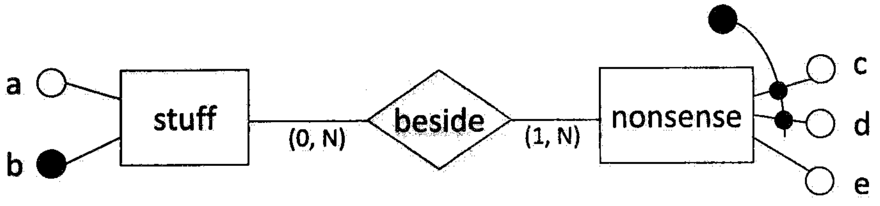
Explain:

Part (c) [1 MARK] Write a key for R in the box below.

Part (d) [2 MARKS] Justify why you know that this is a key.

Question 11. [3 MARKS]

Consider the following ER diagram:

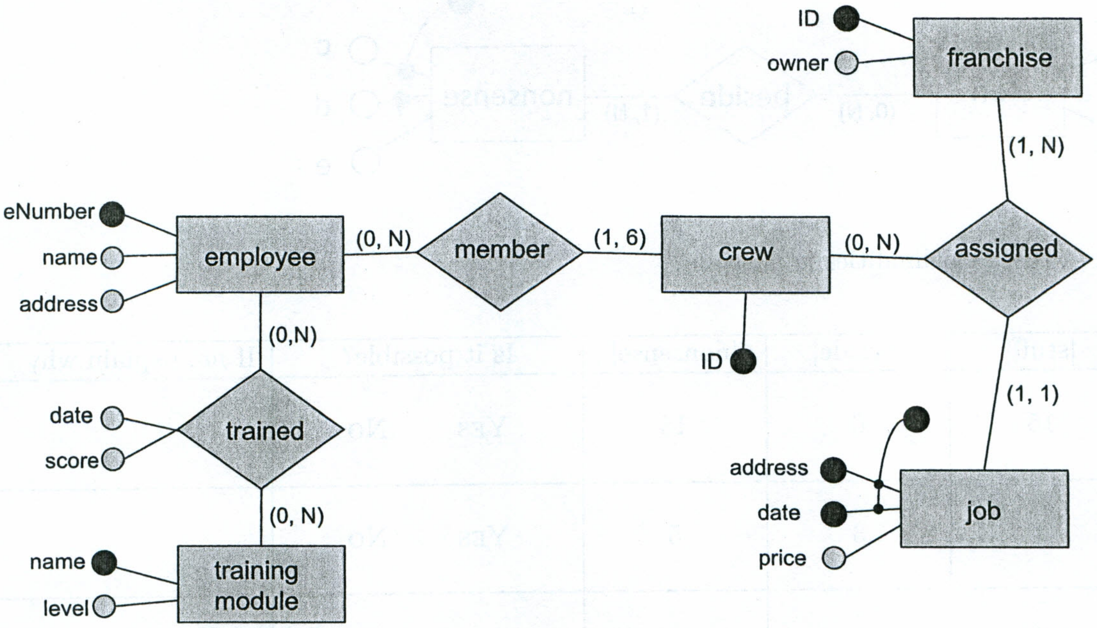


Which of these cardinalities is possible?

|stuff|	|beside|	|nonsense|	Is it possible?	If no, explain why
15	0	15	YES NO	
2	3	5	YES NO	
9	5	2	YES NO	
8	8	8	YES NO	
4	16	4	YES NO	
11	11	5	YES NO	

Question 12. [4 MARKS]

Below is an Entity-Relationship diagram about a company that offers window-washing services to home owners, and has many franchises across the Greater Toronto Area.



The diagram may or may not represent the domain well. Regardless, which of the following is true, according to this Entity-Relationship diagram?

1. The *trained* relationship is a many-to-many relationship.

   True False

2. The *assigned* relationship is a one-to-one relationship.

   True False

3. This model contains a weak entity set.

   True False

4. An employee can be a member of at most 6 crews.

   True False

5. There cannot be two different prices associated with one address.

   True False

6. A franchise cannot have two owners.

   True False

7. One owner can own at most a single franchise.

   True False

8. Two different employees cannot be trained on the same module unless they are trained on different dates.

   True False

Question 13. [8 MARKS]

Translate the Entity-Relationship diagram from the previous question into a relational schema. For each relation, provide its name, attributes and keys. To indicate a key, underline all attributes that are part of the key using a single line. To define referential integrity constraints, use relational notation (that is, using $\subseteq$, not SQL notation).

Question 14. [7 MARKS]

Assume that the following (syntactically correct) DTD is defined in file bookstore.dtd

```
<!ELEMENT book (author+, rating*)>
   <!ATTLIST book isbn ID #REQUIRED>
   <!ATTLIST book title CDATA #REQUIRED>
   <!ATTLIST book year CDATA #REQUIRED>
   <!ATTLIST book type CDATA #IMPLIED>
<!ELEMENT author (#PCDATA)>
<!ELEMENT rating (#PCDATA)>
   <!ATTLIST rating stars CDATA #REQUIRED>
```

Part (a) [4 MARKS]

For each XML document below, the first two lines are syntactically correct. Circle the correct answer to indicate whether the entire document is valid or not. If it is not valid, **circle each error and explain** why it is an error.

1.
```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE book SYSTEM "bookstore.dtd">
<book isbn="n12345" title="The Bone Clocks" type="paperback">
    <rating stars="5">
        Perfection
    </rating>
    <rating stars="4">
        A fabulous read. Knocked my socks off!
    </rating>
</book>
```

Valid Not Valid

2.
```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE book SYSTEM "bookstore.dtd">
<book isbn="54321" title="Each Peach" year="1984" type="childrens" type="hard cover">
    <author>Janet Ahlberg</author>
    <author>Allan Ahlberg</author>
</book>
```

Valid Not Valid

3.
```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE book SYSTEM "bookstore.dtd">
<book isbn="n12345" title="The Bone Clocks" year="2014">
    <author>David Mitchell</author>
</book>
```

Valid Not Valid

4.
```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE book SYSTEM "bookstore.dtd">
<book isbn="n12345" title="The Bone Clocks" year="2014" type="paperback">
    <rating stars="5"/>
    <rating stars="4">
        A fabulous read. Knocked my socks off!
    </rating>
    <author>David Mitchell</author>
</book>
```

Valid Not Valid

Part (b) [3 MARKS]

Suppose we want to be able to validate an XML document containing a list of one or more book elements, nested inside a booklist element. Revise the original DTD, copied below, to allow this.

```
<!ELEMENT book (author+, rating*)>

   ... ATLIST definitions for element "book" omitted ...

<!ELEMENT author (#PCDATA)>

<!ELEMENT rating (#PCDATA)>

   ... ATLIST definitions for element "rating" omitted ...
```

Revise the example instance document below so that it will validate successfully against the revised DTD. Assume that the omitted parts (shown as "...") are correct.

```
<?xml version="1.0" standalone="no"?>

<!DOCTYPE book SYSTEM "bookstore.dtd">

<book ...>
   ...
</book>

<book ...>
   ...
</book>
```

Question 15. [5 MARKS]

Consider the JSON schema below.

```
{"type": "object",
  "properties": {
    "prof": { "type": "string" },
    "num": { "type": "number" },
    "students": {
      "type": "array",
      "items": {
         "type": "object", "properties": {
                          "name": { "type": "string" },
                          "grade": {
                             "anyOf": [
                                { "type": "string", "enum": ["NGA", "WDR"] },
                                { "type": "number"}]
                          }
         },
      "required": ["name", "grade"], "additionalProperties": false
      }
    }
  },
  "required": ["num", "students"], "additionalProperties": true
}
```

For each JSON object below, circle one option to indicate whether or not it is a valid JSON object, and if it is a valid JSON object, whether or not it validates against this particular schema. Briefly explain your answer if you circle either of the first two options.

1.
```
{ "prof": "Craig", "num": 343, "breadth": false,
  "students": [{"name":"Pankaj", "grade": 78},
               {"name":"David",  "grade": "WDR"}]
}
```
Circle one: Invalid JSON Valid JSON but does not validate Valid JSON that validates

Explanation:

2. `{ }`

Circle one: Invalid JSON Valid JSON but does not validate Valid JSON that validates

Explanation:

3.
```
{ "num": 343,
  "students": [{"name":"Pankaj", "grade": 78},
               {"name":"David",  "grade": "WDR", "dropped":true }]
}
```

Circle one: Invalid JSON Valid JSON but does not validate Valid JSON that validates

Explanation:

4.
```
{ "num": 104,
  "students": [],
  "breadth": true
}
```

Circle one: Invalid JSON Valid JSON but does not validate Valid JSON that validates

Explanation:

5.
```
{ [{"name":"Pankaj", "grade": 78},
  {"name":"David",  "grade": "WDR"}] : "students",
  343 : "num"
}
```

Circle one: Invalid JSON Valid JSON but does not validate Valid JSON that validates

Explanation:

Use this page for rough work. If you want work on this page to be marked, please indicate this clearly *at the location of the original question.*

Use this page for rough work. If you want work on this page to be marked, please indicate this clearly *at the location of the original question.*

Use this page for rough work. If you want work on this page to be marked, please indicate this clearly *at the location of the original question.*

Total Marks = 95