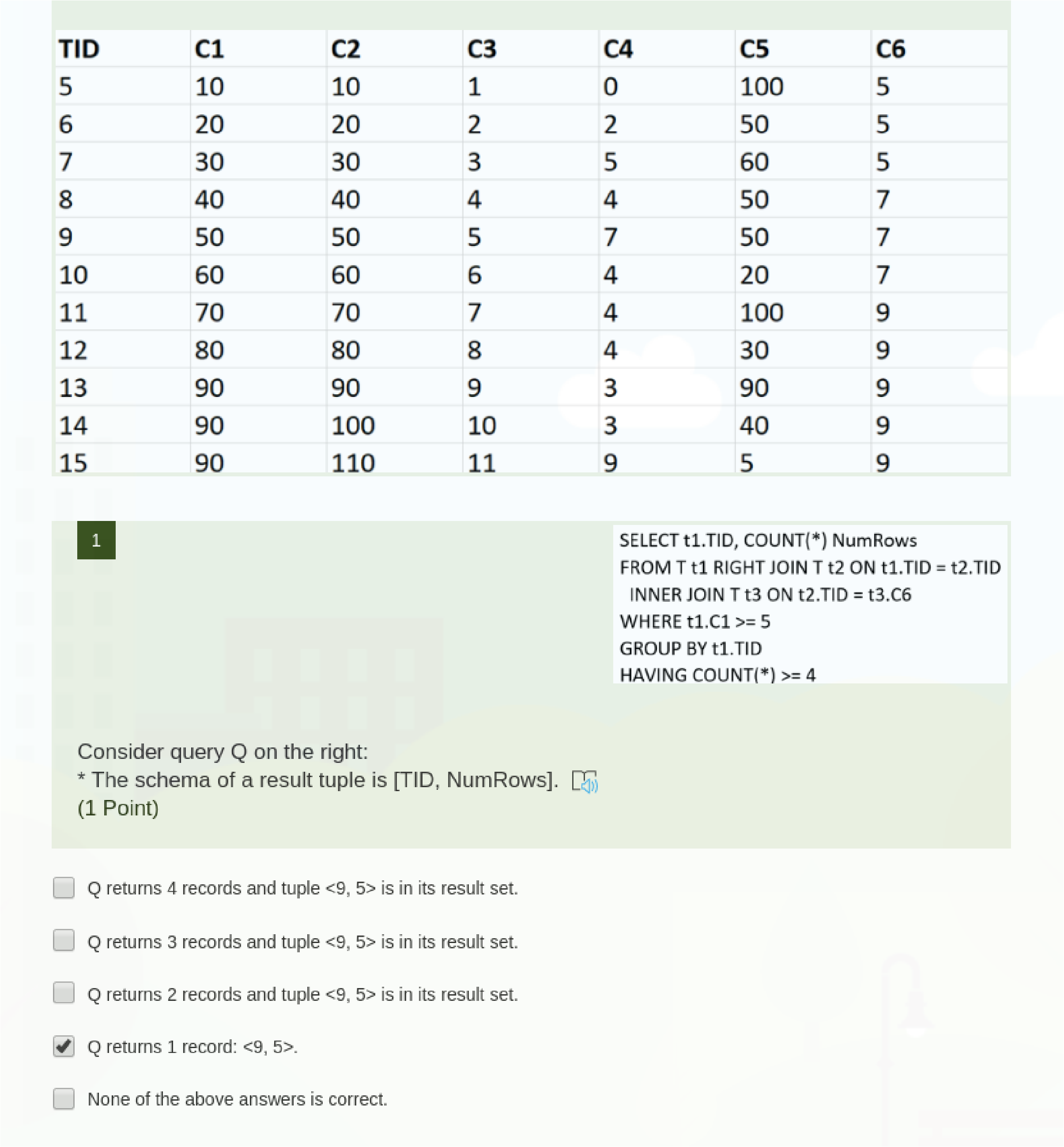


TID	C1	C2	C3	C4	C5	C6
5	10	10	1	0	100	5
6	20	20	2	2	50	5
7	30	30	3	5	60	5
8	40	40	4	4	50	7
9	50	50	5	7	50	7
10	60	60	6	4	20	7
11	70	70	7	4	100	9
12	80	80	8	4	30	9
13	90	90	9	3	90	9
14	90	100	10	3	40	9
15	90	110	11	9	5	9

1

```
SELECT t1.TID, COUNT(*) NumRows
FROM T t1 RIGHT JOIN T t2 ON t1.TID = t2.TID
  INNER JOIN T t3 ON t2.TID = t3.C6
WHERE t1.C1 >= 5
GROUP BY t1.TID
HAVING COUNT(*) >= 4
```

Consider query Q on the right:
* The schema of a result tuple is [TID, NumRows].
(1 Point)

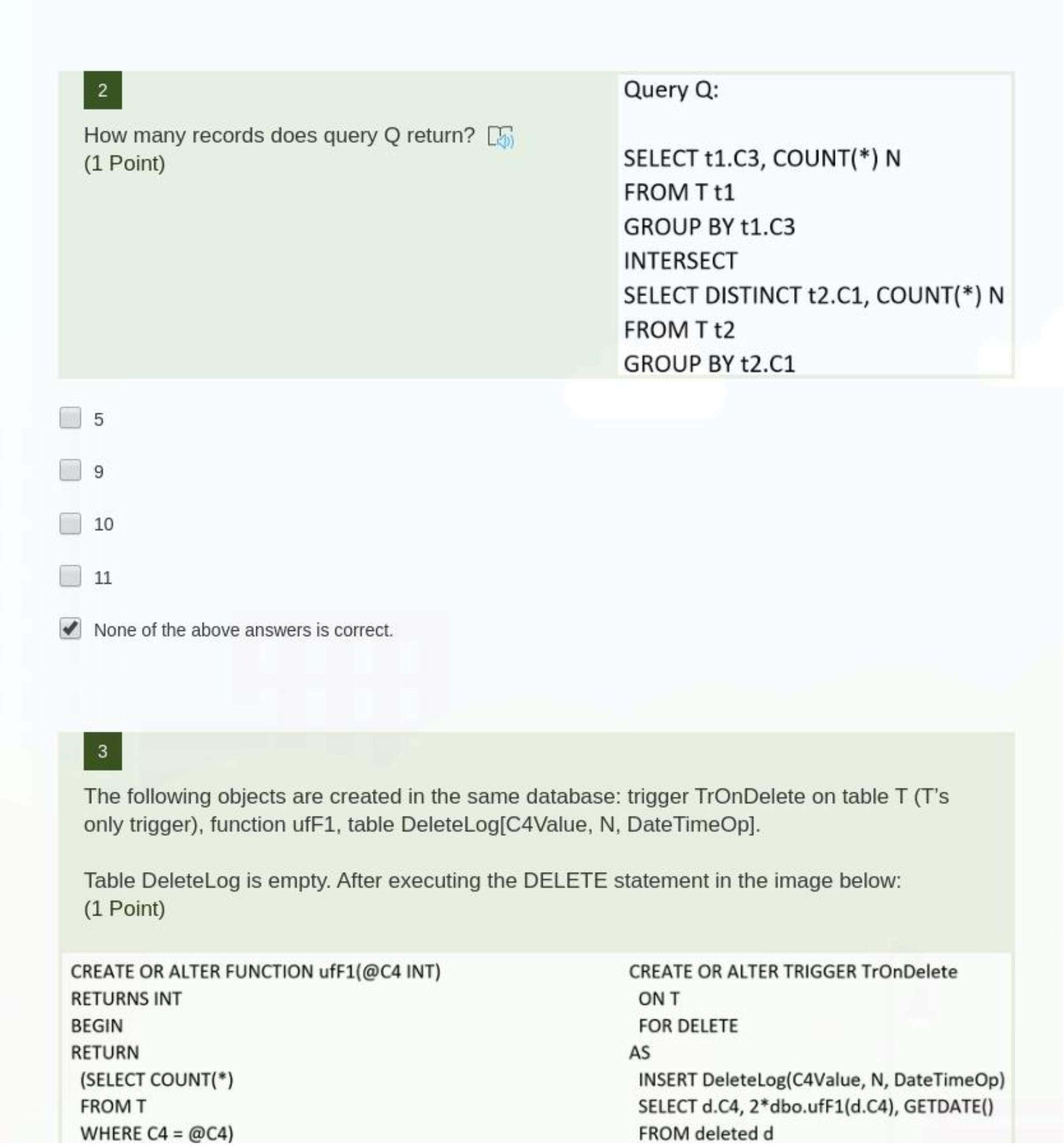
☐ Q returns 4 records and tuple <9, 5> is in its result set.

☐ Q returns 3 records and tuple <9, 5> is in its result set.

☐ Q returns 2 records and tuple <9, 5> is in its result set.

☑ Q returns 1 record: <9, 5>.

☐ None of the above answers is correct.



2

How many records does query Q return?
(1 Point)

Query Q:

```
SELECT t1.C3, COUNT(*) N
FROM T t1
GROUP BY t1.C3
INTERSECT
SELECT DISTINCT t2.C1, COUNT(*) N
FROM T t2
GROUP BY t2.C1
```

□ 5

□ 9

□ 10

□ 11

☑ None of the above answers is correct.

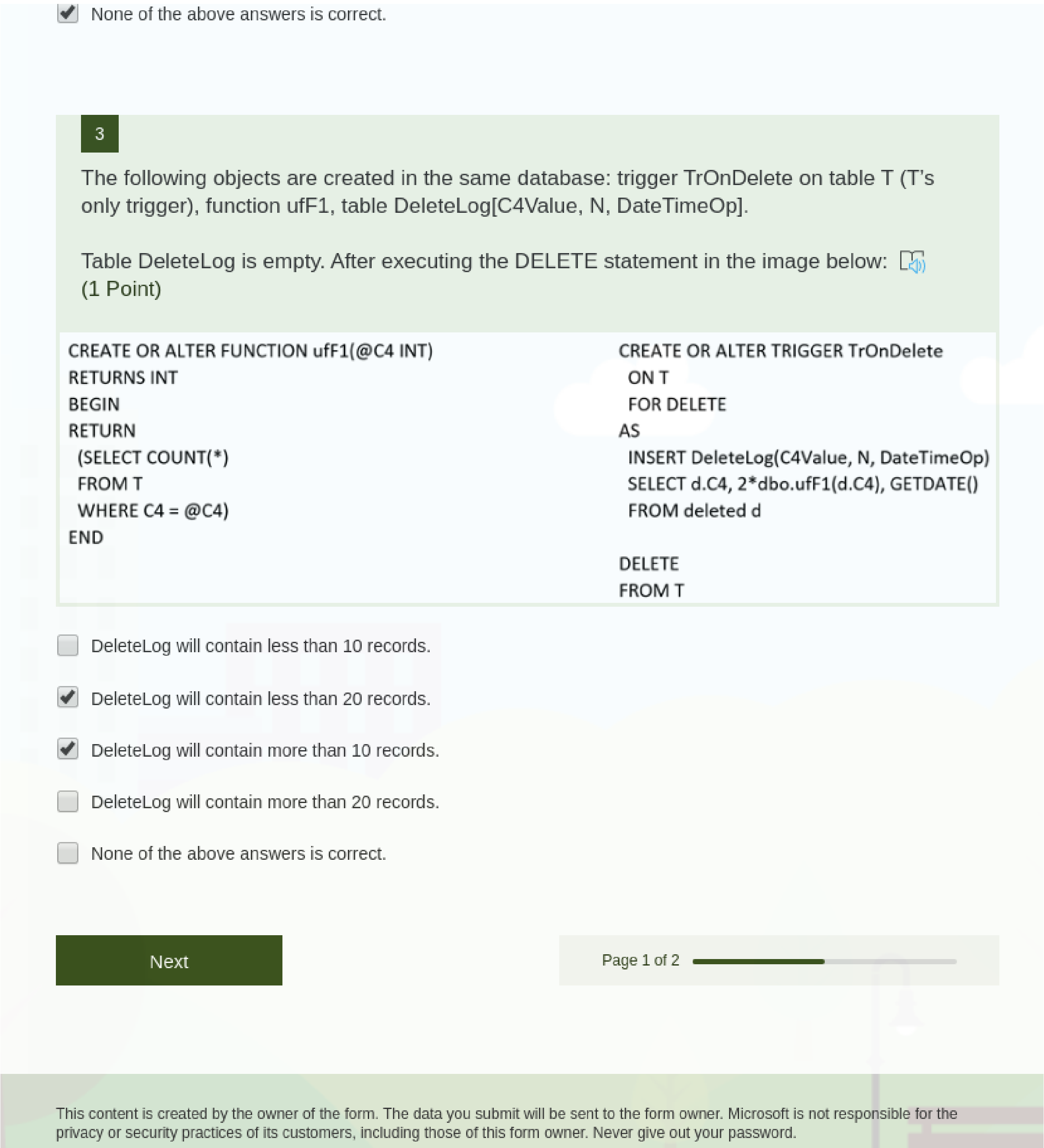
3

The following objects are created in the same database: trigger TrOnDelete on table T (T's only trigger), function ufF1, table DeleteLog[C4Value, N, DateTimeOp].

Table DeleteLog is empty. After executing the DELETE statement in the image below:
(1 Point)

```
CREATE OR ALTER FUNCTION ufF1(@C4 INT)
RETURNS INT
BEGIN
RETURN
 (SELECT COUNT(*)
 FROM T
 WHERE C4 = @C4)
```

```
CREATE OR ALTER TRIGGER TrOnDelete
 ON T
 FOR DELETE
AS
 INSERT DeleteLog(C4Value, N, DateTimeOp)
 SELECT d.C4, 2*dbo.ufF1(d.C4), GETDATE()
 FROM deleted d
```



☑ None of the above answers is correct.

3

The following objects are created in the same database: trigger TrOnDelete on table T (T's only trigger), function ufF1, table DeleteLog[C4Value, N, DateTimeOp].

Table DeleteLog is empty. After executing the DELETE statement in the image below:
(1 Point)

```
CREATE OR ALTER FUNCTION ufF1(@C4 INT)
RETURNS INT
BEGIN
RETURN
 (SELECT COUNT(*)
 FROM T
 WHERE C4 = @C4)
END
```

```
CREATE OR ALTER TRIGGER TrOnDelete
 ON T
 FOR DELETE
AS
 INSERT DeleteLog(C4Value, N, DateTimeOp)
 SELECT d.C4, 2*dbo.ufF1(d.C4), GETDATE()
 FROM deleted d

DELETE
FROM T
```

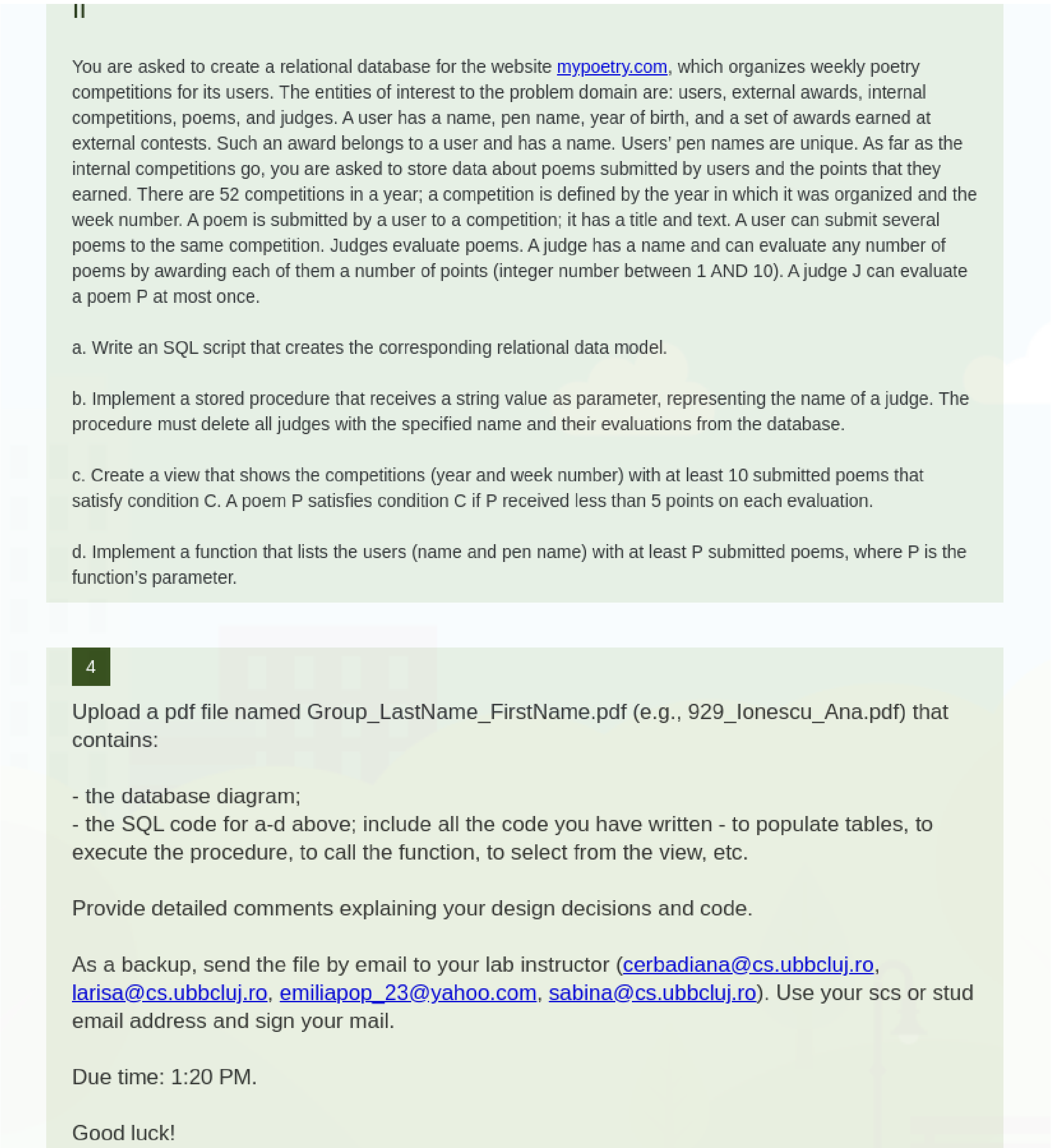
☐ DeleteLog will contain less than 10 records.

☑ DeleteLog will contain less than 20 records.

☑ DeleteLog will contain more than 10 records.

☐ DeleteLog will contain more than 20 records.

☐ None of the above answers is correct.



II

You are asked to create a relational database for the website mypoetry.com, which organizes weekly poetry competitions for its users. The entities of interest to the problem domain are: users, external awards, internal competitions, poems, and judges. A user has a name, pen name, year of birth, and a set of awards earned at external contests. Such an award belongs to a user and has a name. Users' pen names are unique. As far as the internal competitions go, you are asked to store data about poems submitted by users and the points that they earned. There are 52 competitions in a year; a competition is defined by the year in which it was organized and the week number. A poem is submitted by a user to a competition; it has a title and text. A user can submit several poems to the same competition. Judges evaluate poems. A judge has a name and can evaluate any number of poems by awarding each of them a number of points (integer number between 1 AND 10). A judge J can evaluate a poem P at most once.

a. Write an SQL script that creates the corresponding relational data model.

b. Implement a stored procedure that receives a string value as parameter, representing the name of a judge. The procedure must delete all judges with the specified name and their evaluations from the database.

c. Create a view that shows the competitions (year and week number) with at least 10 submitted poems that satisfy condition C. A poem P satisfies condition C if P received less than 5 points on each evaluation.

d. Implement a function that lists the users (name and pen name) with at least P submitted poems, where P is the function's parameter.

4

Upload a pdf file named Group_LastName_FirstName.pdf (e.g., 929_Ionescu_Ana.pdf) that contains:

- the database diagram;
- the SQL code for a-d above; include all the code you have written - to populate tables, to execute the procedure, to call the function, to select from the view, etc.

Provide detailed comments explaining your design decisions and code.

As a backup, send the file by email to your lab instructor (cerbadiana@cs.ubbcluj.ro, larisa@cs.ubbcluj.ro, emiliapop_23@yahoo.com, sabina@cs.ubbcluj.ro). Use your scs or stud email address and sign your mail.

Due time: 1:20 PM.

Good luck!