1.264 Final Exam
Fall, 2001

Exam guidelines:

1. Three hours are allowed to complete the exam.
2. If there are questions or clarifications, ask the proctor.
3. Write your answers in the examination booklet provided.
4. Open notes.
5. No books allowed.
6. No laptop computers are allowed.
7. Calculators are not necessary and are not allowed.
8. 101 points total (generally 20 points for each of 5 questions).
9. Put your name and email address on the exam booklet.
10. Short answer questions: Your answers are limited to a maximum of 2-4 sentences or phrases. Two sentences or phrases are completely adequate to answer any of these questions; 4 sentences are not necessary. Demonstrate that you understand the principles and key points. You will receive full credit for an answer if you make the principal observation(s) that the question is asking for. Details are not necessary.

1. Software process. (20 points)

Chris was in charge of version 1 of MIT's chemical inventory system, MITCIS. With higher security required these days, MIT realized that its informal system of ordering, delivering and disposing of chemicals had to change. Chris had a general idea of the capabilities necessary from having taken 1.264 and from the first meeting of the MIT steering committee for the project. Dave was the head of the committee. "Chris, how long is MITCIS going to take, he asked?"

"I think it will take about 9 months, but that's just a rough estimate at this point," Chris said.

"That's not going to work," Dave said. "I was really hoping you'd say 3 or 4 months. We absolutely must have MITCIS within 6 months; we can't tolerate our security exposure any longer than that, at the absolute worst. You know how serious this problem is, don't you? Can you do it in 6 months?"

"I'm not sure," Chris said honestly. "I'd have to look at the project more carefully, but I can try to find a way to get it done in 6."

"Treat 6 months as a goal, then," Dave said. "That's what it's got to be, anyway." The rest of the committee agreed.

By week 5, additional work on the system requirements had convinced Chris that the project would take closer to the initial 9-month guess than to 6 months, but he thought that with some luck he might still be able to complete it in 7 or 8 months. He didn't want to be labeled a troublemaker or uncooperative, and he wanted to stay in the good graces of the committee, since it would heavily influence his promotions and future assignments.

Chris' team made steady progress but requirements analysis took longer than they had hoped. Every lab and department had a different way of doing things, and it took a long time to reach agreement on a common approach. They were now almost 3 months into what was supposed to be a 6-month project. "There's no way we can do the rest of the work we have to do in 3 months," he told Dave. He told Dave he needed a 2-month extension and rescheduled the project to take 8 months. He felt it could be done in 9 or 10 months, realistically, but he didn't feel he could present those times to the committee.

A few weeks later, Chris realized that the designs for the database, Web page and program weren't proceeding as quickly as he had hoped either. "Implement the parts you can do easily," he told the team. "We'll worry about the rest of the parts when we get to them."

Chris met with the MIT steering committee. "We're now 7 months into our 8-month project. Detailed design is almost complete, and we're making good progress but we can't complete the project in 8 months." Dave announced his second schedule slip to 10 months. Dave grumbled and asked Carl to find a way to bring the schedule back to about 8 months.

At the 9-month mark, the team had completed detailed design, but program development had still not begun on some parts of the system. It was clear that the team couldn't make the 10-month schedule either. Chris announced the third schedule slip, to 12 months. Dave's face turned red when Chris announced the slip, and the pressure from the MIT committee became more intense. Chris began to feel his job was on the line.

Coding proceeded fairly well, but a few areas needed redesign and recoding. The team hadn't coordinated design details among the Web page, database and Visual Basic sub-teams well, and some of the implementations conflicted. At the 11-month steering committee meeting, Chris announced the fourth schedule slip, to 14 months. Dave became livid. "Do you have any idea of what you're doing?" he yelled. "You obviously don't have any idea when the project is going to be done! I'll tell you when the project will be done! It's going to be done by the 12-month mark, or you're going to be out of a job! You and your team are going to work 80 hours per week until you deliver!" Chris felt his blood pressure rise, especially since Dave had backed him into an unrealistic schedule in the first place. But he knew that with four schedule slips, he had no credibility left.

Chris told his team about the meeting. They worked hard and managed to deliver the software in just over 13 months.

a. List at least 5 errors made by this development team in their execution of this project. (7 points)

1. No resource estimates (function points, lines of code, schedule months, person months) were made at the start of the project. Chris gave an off-the-cuff estimate when asked at the first meeting.
2. No uncertainty was communicated to the steering committee. Chris gave a point estimate, and he didn't state any assumptions on the size of the project, complexity of requirements, etc.
3. No discussions of tradeoffs in functions or team size to meet the 6-month date were held. Chris should have done this as requirements became complex, and even at the start of the project.
4. Chris did not communicate the true schedule as the project progressed. The real estimates were not given, and no proactive information was given before there was a

problem. If requirements took twice as long as expected, so would every other part of the project, and this was not addressed.
5. The team did not use the spiral model (or other appropriate model such as evolutionary prototyping, etc.). They used a waterfall, which is inappropriate for rapid development, which this project clearly required.
6. The design was not coordinated properly since the pieces did not fit together. This is poor management and possibly poor use of tools.
7. Integration was not begun until too late in design and coding. Leaving the hard parts until the end is a mistake; they won't integrate properly.

b. Outline the key steps in the resource estimation that the team should have used to avoid these errors/problems. List at least 5 steps; describe each in 1-2 sentences or phrases. (6 points)

1. Quickly estimate the number of Web pages (or VB screens) and database tables and external interfaces, if any, to be built.
2. Estimate the function points for each.
3. Based on the type of software to be used (database, VB, etc.) for each part of the system, estimate the lines of code.
4. Look up the schedule and person months for the appropriate project type (business) and software maturity (nominal, at best).
5. Apply a range of uncertainty based on the stage of the project.

c. Outline the key steps in the software development process that the team should have used to avoid these errors/problems. List at least 5 steps; describe each in 1-2 sentences or phrases. (7 points)

1. Use the spiral model (or other evolutionary model)
2. Use 3 month spirals to ensure that a system can be shipped at the 3 month point (a very simple one, but this ensures integration) and at the 6 month point (also likely to be very simple)
3. Do requirements in the first 3 weeks of spiral 1, instead of letting them go for 3 months. You can start coding with a rough set of requirements for the first spiral, since you'll be finding all sorts of other problems with the data, pages, interfaces, etc. Requirements should be roughly ¼ of the effort in one spiral

4. Do design in the next 3 weeks. It should also be ¼ of each spiral. Design went on for 6 months (!!) in this project, as long as the total project was supposed to take.
5. Build an initial prototype, similar to what you did in 1.264, in about 4-5 weeks. Make sure everything integrates. You can do better than in 1.264 because you have more time and more knowledge.
6. Test, review and debug for 2 weeks or so in spiral 1 and show an initial product to the steering committee at the 3-month mark. If it takes until the 4-month mark, that's not a major problem, though it will tell you that the product won't be complete until 7 months, since 3 month spirals are the shortest possible.
7. The committee, if it sees a working prototype at 3-4 months, will be more supportive, can help limit requirements changes, etc.

2. **Data model (20 points)**

   You are given the following set of business rules for MIT's new chemical information system, MITCIS.

   a. MITCIS supports 150 laboratories at MIT in departments, research centers and administration (support). Information is maintained for lab name, name of person authorized to order chemicals, and his or her phone number, MIT address and email. Each lab has a unique ID. Each lab is in one of 5 'regions' at MIT based on its location; the regions are named Central, East, West, North and Northwest.
   b. Each lab has an owning organization (department, research center or administration). MIT has tables of departments, research centers and administrative departments. Each of the three tables has some elements in common: a unique ID, name, address and phone number. They also have unique attributes:
      i. Departments have the MIT school to which they belong
      ii. Research centers have credit limits
      iii. Administrative departments require a second person to authorize each purchase, whose name, address and email are listed.
   c. There are two locations where chemical inventory is kept at MIT. One serves Central and West. The other serves Central, East, North and Northwest. The locations are called A and B. You don't need to model their inventory level.
   d. MIT has a chemical product distribution workforce. All are MIT employees. Each is assigned to one region. The name, employee ID, email and address of each employee are recorded.
   e. A lab is served by at least one support person and usually by multiple support persons. Each employee supporting a lab has a role: primary support, backup support or specialist.
   f. Labs enter orders by phone or through a Web page. Each order has a unique number and order date. Each is assigned to a chemical support person, who tracks its status.
   g. The order can have many items. Each line has an item ID (which is a chemical product ID) and a quantity.

There are other tables containing chemical products, etc. but you don't need to model them. Note that there is more than one indirect relationship between lab and employee, which would cause major complications and confusion in a real system like MITCIS, but which is frequently present.

You must draw a data model that corresponds to this set of business rules. Follow these steps. You only need to turn in one drawing that includes all the elements listed in steps a-e.

a. **Draw a box for each entity: give each an appropriate name (5 points)**

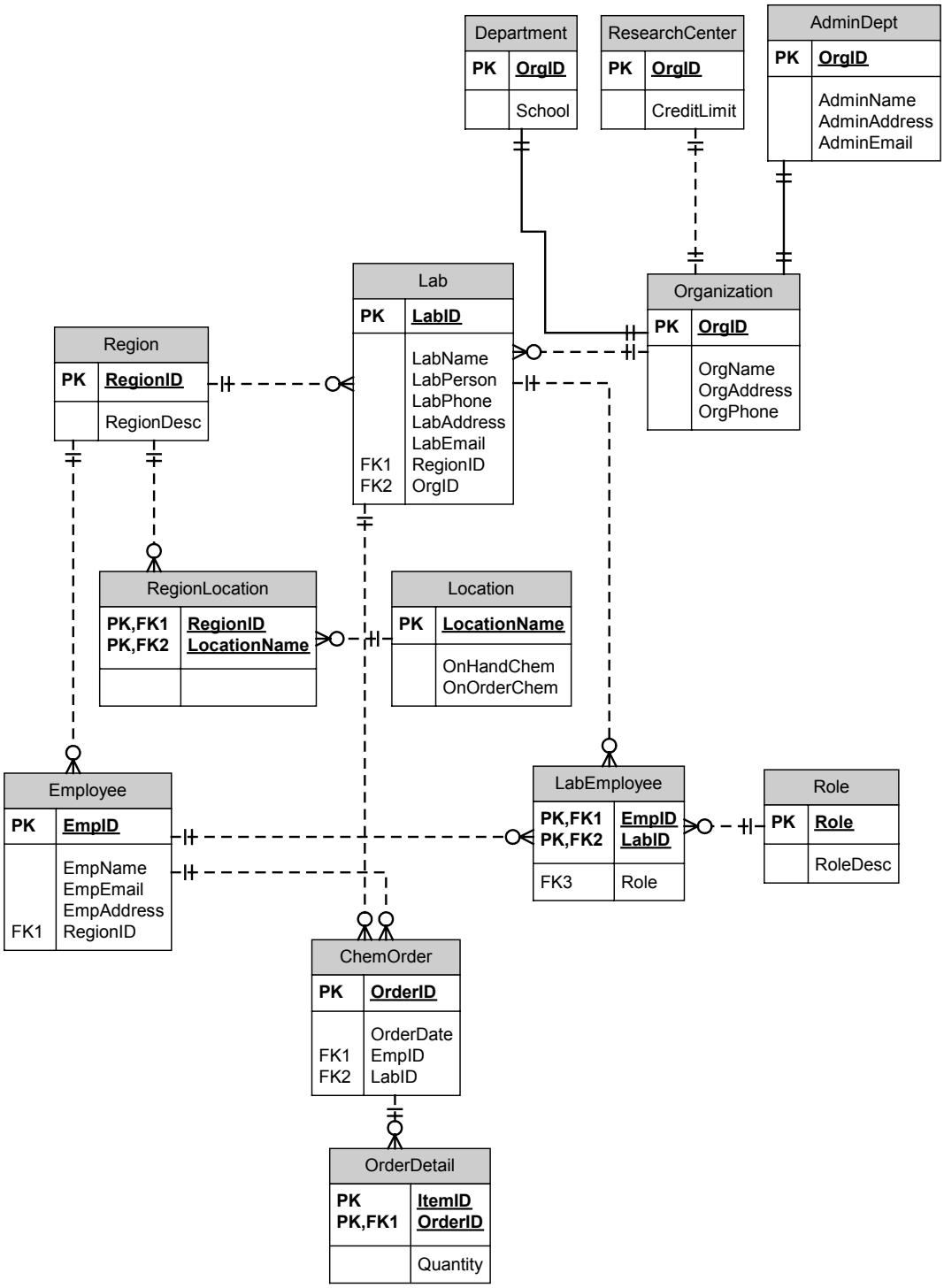
b. **List the attributes in the box for each entity (5 points)**

c. **Indicate the primary key for each entity by placing the phrase (PK) next to its name. (2 points)**

d. **Draw all relationships between the entities in the model. Indicate foreign keys by placing the phrase (FK) next to attributes that are foreign keys. (5 points)**

e. **Indicate the cardinality of the relationship: many-many, many-one or one-one. Use crows-foot notation; if you use another notation, define it. (3 points)**

To repeat: You only need to turn in one drawing that includes all the elements listed in steps a-e above.


Department
PK OrgID
School
ResearchCenter
PK OrgID
CreditLimit
AdminDept
PK OrgID
AdminName
AdminAddress
AdminEmail
Lab
PK LabID
LabName
LabPerson
LabPhone
LabAddress
LabEmail
FK1 RegionID
FK2 OrgID
Organization
PK OrgID
OrgName
OrgAddress
OrgPhone
Region
PK RegionID
RegionDesc
RegionLocation
PK,FK1 RegionID
PK,FK2 LocationName
Location
PK LocationName
OnHandChem
OnOrderChem
Employee
PK EmpID
EmpName
EmpEmail
EmpAddress
FK1 RegionID
LabEmployee
PK,FK1 EmpID
PK,FK2 LabID
FK3 Role
Role
PK Role
RoleDesc
ChemOrder
PK OrderID
OrderDate
FK1 EmpID
FK2 LabID
OrderDetail
PK ItemID
PK,FK1 OrderID
Quantity

3. **Database. (21 points)**

Based on the data model in question 2, you construct a database where the tables, attributes and relationships correspond exactly to the data model structure. You are now asked to write the following SQL queries against your database. You may use either 'standard' SQL or MS Access SQL syntax. Indicate which you are using. Briefly define any variable you use that isn't obvious.

Do any three of the following four questions (7 points each). If you wish, you may do all 4; you will receive credit for the best three answers (and zero points for the fourth one).

a. **List the name of the administrative departments that have labs served by the West region.**

Standard SQL:

```
SELECT AdminDept.AdminName
FROM AdminDept, Organization, Lab, Region
WHERE AdminDept.OrgID=Organization.OrgID AND Organization.OrgID=
Lab.OrgID AND Lab.RegionID= Region.RegionID AND Region.RegionID="West"
```

Access SQL:

```
SELECT AdminDept.AdminName
FROM Region INNER JOIN ((Organization INNER JOIN AdminDept ON
Organization.OrgID = AdminDept.OrgID) INNER JOIN Lab ON (AdminDept.OrgID =
Lab.OrgID) AND (Organization.OrgID = Lab.OrgID)) ON Region.RegionID =
Lab.RegionID
WHERE (((Region.RegionID)="West"));
```

b. **List the laboratory names that each employee supports, for all employees. Display the employee ID, employee role and lab name. Order by employee ID.**

Standard SQL:

```
SELECT LabEmployee.EmpID, LabEmployee.Role, Lab.LabName
FROM Lab, LabEmployee
WHERE Lab.LabID= LabEmployee.LabID
ORDER BY LabEmployee.EmpID;
```

Access SQL:

```
SELECT LabEmployee.EmpID, LabEmployee.Role, Lab.LabName
FROM Lab INNER JOIN LabEmployee ON Lab.LabID = LabEmployee.LabID
ORDER BY LabEmployee.EmpID;
```

c. **List the item number with the maximum quantity in each order. List the item number, item quantity and order number. (Ignore the Access**

restriction that only the single column with the max operation can be displayed in the result set, if you use Access SQL syntax.)

Standard SQL:

```
SELECT Max(OrderDetail.Quantity), OrderDetail.OrderID, OrderDetail.ItemID
FROM OrderDetail
GROUP BY OrderDetail.OrderID;
```

Access SQL:

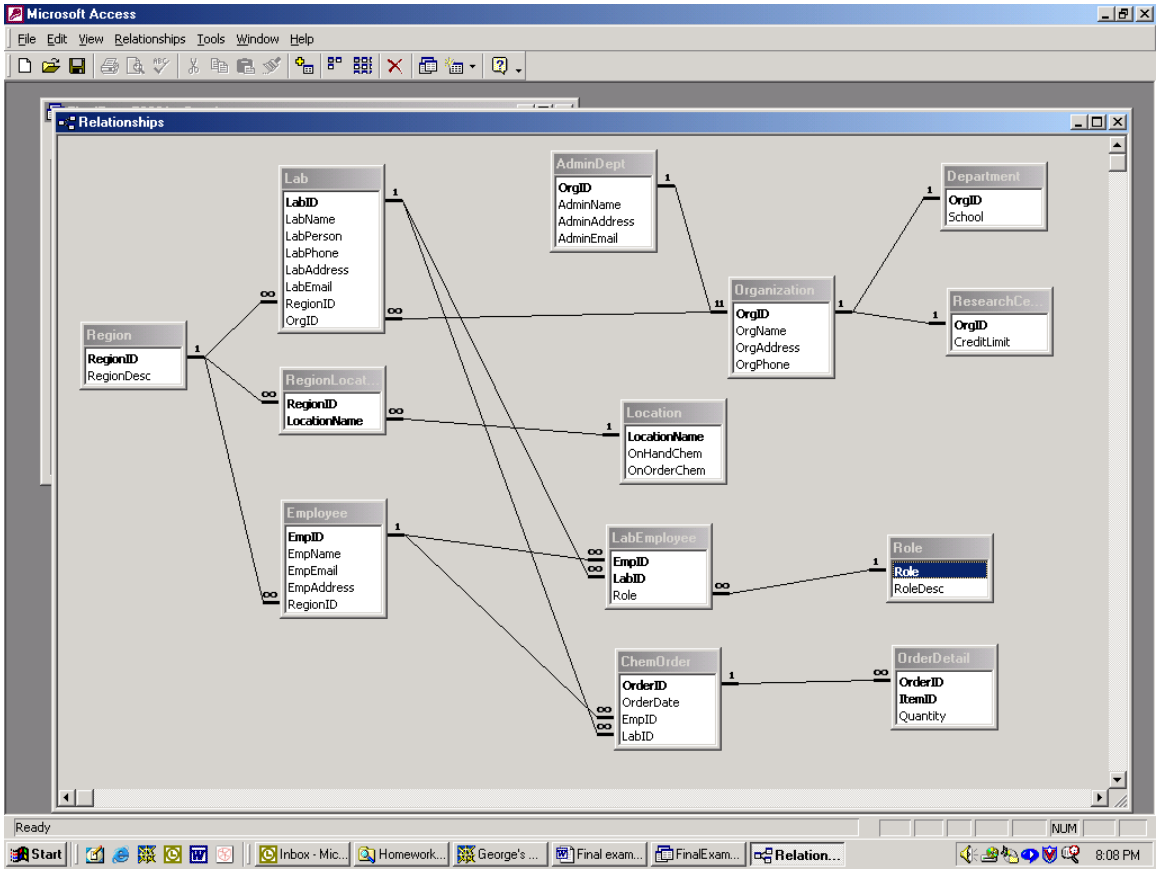
```
SELECT Max(OrderDetail.Quantity), OrderDetail.OrderID, OrderDetail.ItemID
FROM OrderDetail
GROUP BY OrderDetail.OrderID;
```

d. Delete employees that are supporting no laboratories (i.e., whose employee IDs are not present in the laboratory-employee table). (Assume that these employees are not in any orders that would prevent them from being removed from the employee table.)

Standard SQL, Access SQL:

```
DELETE Employee.*
FROM Employee
WHERE Employee.EmpID NOT IN (SELECT LabEmployee.EmpID FROM
LabEmployee);
```

To repeat: do only 3 of the 4 questions above.


Microsoft Access
File Edit View Relationships Tools Window Help
Relationships
Lab
LabID
LabName
LabPerson
LabPhone
LabAddress
LabEmail
RegionID
OrgID
AdminDept
OrgID
AdminName
AdminAddress
AdminEmail
Department
OrgID
School
Organization
OrgID
OrgName
OrgAddress
OrgPhone
ResearchCe...
OrgID
CreditLimit
Region
RegionID
RegionDesc
RegionLocat...
RegionID
LocationName
Location
LocationName
OnHandChem
OnOrderChem
Employee
EmpID
EmpName
EmpEmail
EmpAddress
RegionID
LabEmployee...
EmpID
LabID
Role
Role
Role
RoleDesc
ChemOrder
OrderID
OrderDate
EmpID
LabID
OrderDetail
OrderID
ItemID
Quantity
Ready
NUM
Start
Inbox - Mic...
Homework...
George's ...
Final exam...
FinalExam...
Relation...
8:08 PM

4. **World Wide Web. (20 points)**

For each of the following questions, give a short answer (2-4 sentences or phrases):

a. Does TCP/IP disconnect after each HTTP response and request? Why or why not? (4 points)

Yes, it disconnects after each HTTP response and request. If the session were maintained across responses and requests, it would occupy resources on the server. It would also have be shut down after a timeout period if the user didn't explicitly close it—this is really just an example of the excess resource use if the connection stayed up.

b. Does SSL disconnect after each HTTP response and request? Why or why not? (4 points)

No, it does not. Because the negotiation between client and server is expensive, involving public keys to generate a session key and examining certificates, it is done once at the start of a session. If it disconnected with each request and response, it would be very slow. Adding a complication to SSL would also increase the risk of a bug that would become a security hole.

c. What is the difference between an HTTP POST and GET request? Can data be sent from the browser to the HTTP server with both? (4 points)

A GET request sends a URL and parameters after a ? delimiter. It is intended to 'get' the next Web page, possibly based on the parameters. The blank line terminates the GET request. A POST request is designed to send data to a URL. After the blank line, the server expects to receive all the posted data, so POST is better suited when much data is to be sent. Both GET and POST can often be used interchangeably, though.

d. Can an Active Server Page send a script to be executed on a browser? If so, how does it distinguish between scripts executed on the server versus a script to be executed on the browser? (2 points)

Yes, ASP can send a browser script. In the ASP, scripts that are enclosed in the tags <% …%> or <SERVER> and </SERVER> are executed by the server. Also, server side includes, which are delimited by < -- ! … -- !> execute on the server. Scripts delimited by <SCRIPT> and </SCRIPT> are sent to the browser to be executed there.

e. If an HTML form contained a drawing area where a user could draw a picture with a mouse and then POST the form, and your database could store the image directly (as a binary object) in a table, could an Active Server Page write the image into the database? You may

assume the image is a .gif file. Why or why not? If it's possible, briefly describe how you would do it. (6 points)

Note on part e: In MITCIS you would like the support person to be able to draw a diagram of safety procedures for a chemical showing, for example, the equipment needed if there is a spill, isolation distances, etc. A simple picture is often very effective, and the MITCIS system would like to be able to generate these pictures in a browser and store them on the server.

Answer:
You can post a .gif file using POST. The POST HTTP request would contain the URL of the Active Server Page that can process a .gif. The body of the gif would be sent after the blank line in the HTTP request as a binary file represented as character data. (You've all probably seen attachments that didn't get processed correctly and show up as pages of random characters.) The Active Server Page that receives the POST request would have VBScript or other code to read the .gif line by line, transform it as needed, and write it to a file on the server.

5. **Telecommunications networks. (20 points)**

The MITCIS system will be implemented using MIT's intranet, which has fiber, copper and wireless layer 1 links.

a. **Describe two reasonable ways that MIT can extend its intranet to MIT living groups and offices in downtown Boston, across the Charles River. The distance is about 3 kilometers. Living groups contain about 40 persons each; there are perhaps 15 of them, along with 5 offices. Most are adjacent to other living groups or offices. MIT cannot place any cable of any sort itself because it does not have permission. (4 points)**

DSL: 3 kilometers is 1.9 miles or about 10,000 feet, which is within DSL distance. This is inexpensive broadband and would be appropriate. Separate DSL lines could be provided for each living group and office.
Microwave. Because the distance is short and there are no buildings in the way, and there are buildings on which to place towers, microwave is feasible. To limit the number of transmitter/receivers, Ethernet would be run between adjacent living groups and offices on the Boston side.
Cable modem. Similar to DSL, but because it is shared, security would be a major problem. Traffic would probably need to be encrypted, which would slow it down.
ISDN: Expensive and narrowband, but an option. Not very attractive.

b. **Of the options you defined in part a above, which would you recommend and why? (4 points)**

DSL is likely to be most feasible, since each living group and office can have its own, dedicated, relatively secure broadband connection. DSL is probably the best recommendation.
Microwave is possible but would require several sites in Boston, since MIT can't connect non-adjacent sites with its own cable.
Cable modem is similar to DSL but would require encryption, making its performance slower.

c. **MIT wishes to implement IPSec (secure IP). What layer or layers does this affect? Are any changes required at the other layers? Consider only layers 1, 2, 3, 4 and 7, as we have in class. (4 points)**

IPsec, or secure IP, affects only layer 3, the IP layer. No changes are required in the other layers; the layered model provides exactly this independence and this is exactly why the layered model has been implemented.

d. If MIT implements IPSec as in part c above, what software must it change on clients (PCs, workstations) and servers? (4 points)

MIT would need to change the TCP/IP software on every PC, workstation, server and other device on the network. Strictly speaking, only the IP software would change, but it almost always comes bundled as TCP/IP (including UDP/IP, which we didn't discuss in class).

e. If MIT eliminates all copper layer 1 links and converts them to fiber or wireless in its core network, what software and hardware on clients (PCs, workstations) and servers must be changed? You may assume that PCs will still be connected by 10BaseT or 100BaseT cable to hubs/routers, or by wireless connections to hubs/routers. The core network is what connects bridges, routers and servers to each other and to access points to the Internet. Assume that the protocol is Ethernet over copper, and remains Ethernet over fiber or wireless (4 points)

The layer 2 cards, typically Ethernet, that are currently connected to copper layer 1 links must be changed to cards that work with fiber or wireless links. No software changes are required.