## Structure of _messages.dat file: Example

| | **data**<br>(RECORD_DATA_LEN bytes) | **next**<br>(pointer to next record: 4 byte int) |
|---|---|---|
| **0** | Dear Susan, how are you today? I am going to try to write you more frequently ab | 2 |
| **1** | Hi Johnny, do you want to go to the movies tonight? I have free passes. Suzie | -1 |
| **2** | out the material that we had left open last time. It seems that the entries that | 3 |
| **3** | you had included in the invoices contain some calculation errors. For example, t | 4 |
| **4** | he first item has a price of $39.75 and there were 5 items ordered. The resultin | 6 |
| **5** | To everyone at the sales department: This is a reminder that our monthly luncheo | 8 |
| **6** | g price should be $39.75 x 5 = $198.75 but you have it showing as $201.25, and t | 7 |
| **7** | he tax is calculated based on this figure of $201.25. Could you please review th | 9 |
| **8** | n will take place at Favourite Pizza, 275 Pepperoni Road. See you all there. | -1 |
| **9** | e entire worksheet and email me a new copy. I truly appreciate it. Sincerely Pet | 10 |
| **10** | er, Inventory Manager. | -1 |

### Notes

Left column of the table shows record numbers. They are not part of the messages file.
A message is broken into several records if it takes more than RECORD_DATA_LEN bytes. The rightmost column points to the record where the message continues. The message ends in the record whose **next** has a value of -1 (END_OF_MESSAGE)

### Task

Using the methods of the Record class, create a _message.txt file containing the records and **next** pointers as shown above. This file will be used to test a new set of methods.

**Message** class

Create a class called **Message**
This class will handle the reading, writing and deleting of messages

**variables**
Declare a private string variable **text** and initialize it to a null string

**constructors**
Create a default constructor with **text** and initialize it to a null string

**methods**
Create a method with the signature

```
public void readFromMessagesFile(int recordNumber)
```

that reads an entire message from the messages file

The method does not access the messages file directly but indirectly through the methods of the Record class. The method takes as a parameter the record number where the message begins. It then follows the **next** pointers until an END_OF_MESSAGE is found.

Pseudo-code is below:

```
public void readFromMessagesFile(int recordNumber){
     declare a string variable data and set it to null string
     declare and instantiate a new Record object named record

     do
          read from the messages file the record at recordNumber
          concatenate data with the text (as a string) contained in record
          set recordNumber to next record

     while the recordNumber does not indicate the end of message

     set text to data (this statement will be changed later)
}
```

Create the toString() method

Include code in this method to print the contents of **text** preceeded with a label. For example, when this method is called we would see something like this

```
Message text: Hi Johnny, do you want to go to the movies tonight? I have free passes. Suzie
```

**EmailServer.java**

**test 1**
In the main() method

Declare and instantiate an object of the **Message** class
Make a call to the method readFromMessagesFile() so that the message starting at record 1 is retrieved
Print the contents of the retrieved message

**test 2**
Add code to retrieve and print the message starting at record 0

**test 3**
Add code to retrieve and print the message starting at record 5

Verify that all printed messages are correct