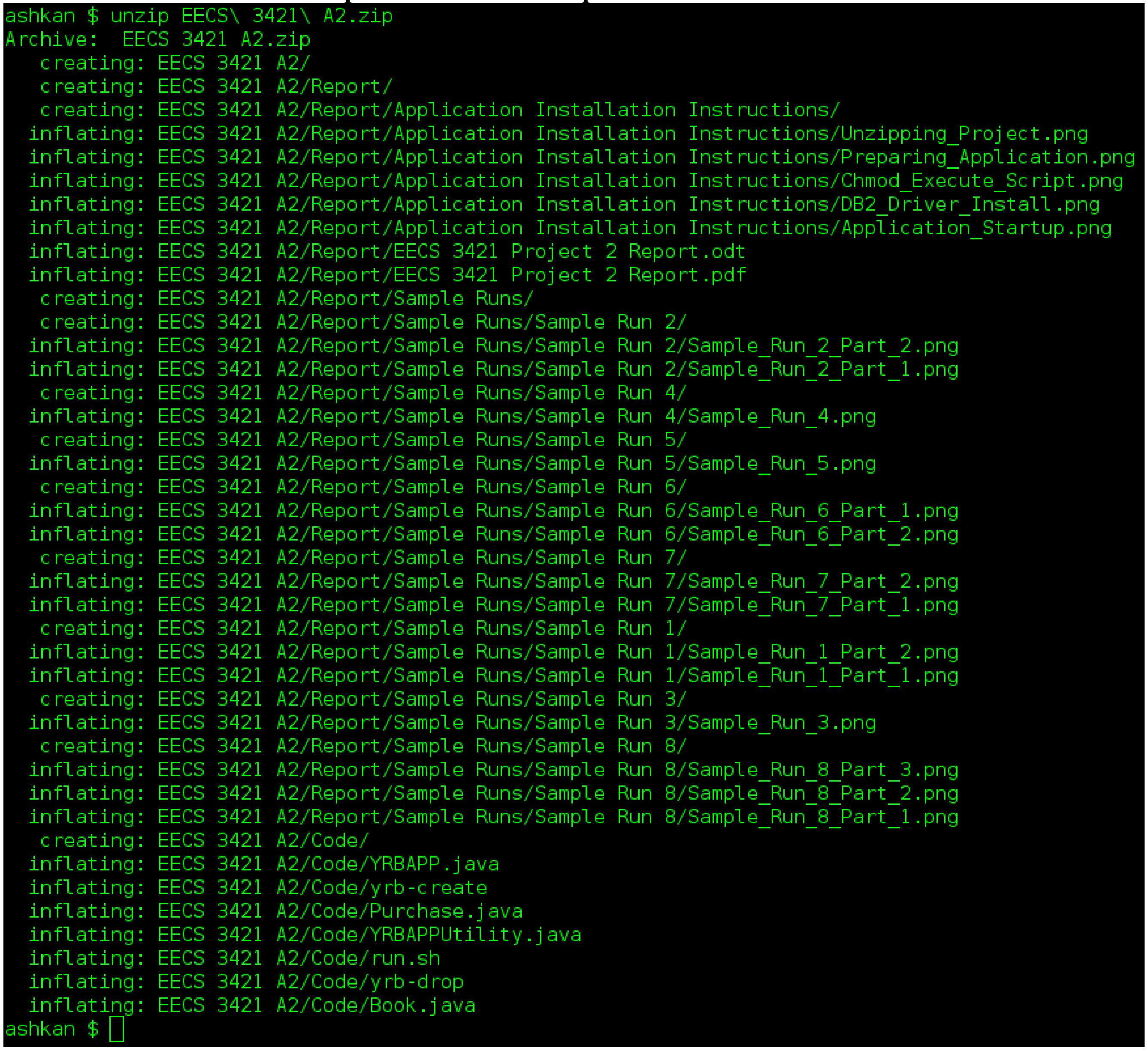
Program User Manual:

Note that it is assumed that the program will be run on the prism computers on the Linux OS.

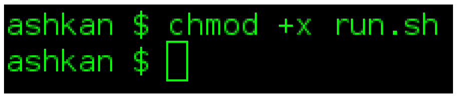
1. First save **EECS 3421 A2.zip** file onto your machine.

2. Unzip it either by right clicking and then choosing **Extract Here** or by entering the following command in the terminal: **unzip EECS\ 3421\ A2.zip**



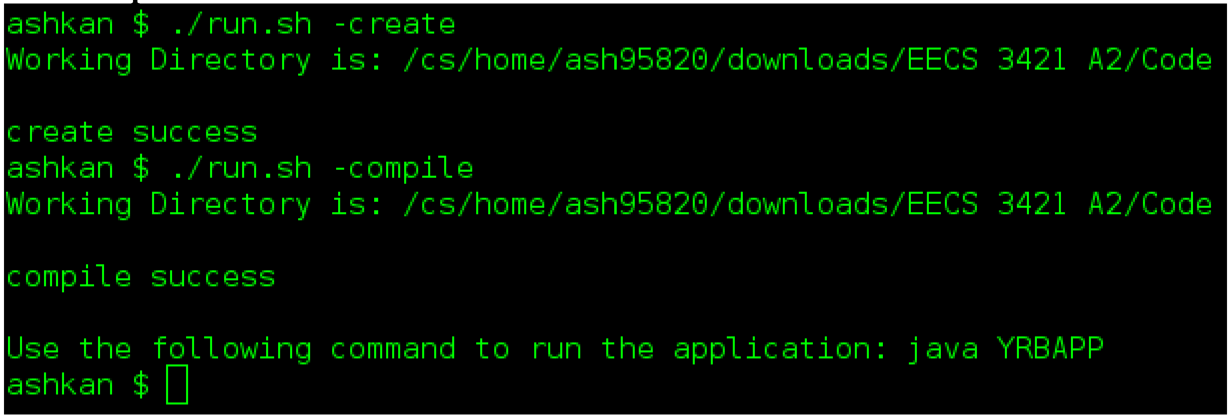
```
ashkan $ unzip EECS\ 3421\ A2.zip
Archive:  EECS 3421 A2.zip
   creating: EECS 3421 A2/
   creating: EECS 3421 A2/Report/
   creating: EECS 3421 A2/Report/Application Installation Instructions/
  inflating: EECS 3421 A2/Report/Application Installation Instructions/Unzipping_Project.png
  inflating: EECS 3421 A2/Report/Application Installation Instructions/Preparing_Application.png
  inflating: EECS 3421 A2/Report/Application Installation Instructions/Chmod_Execute_Script.png
  inflating: EECS 3421 A2/Report/Application Installation Instructions/DB2_Driver_Install.png
  inflating: EECS 3421 A2/Report/Application Installation Instructions/Application_Startup.png
  inflating: EECS 3421 A2/Report/EECS 3421 Project 2 Report.odt
  inflating: EECS 3421 A2/Report/EECS 3421 Project 2 Report.pdf
   creating: EECS 3421 A2/Report/Sample Runs/
   creating: EECS 3421 A2/Report/Sample Runs/Sample Run 2/
  inflating: EECS 3421 A2/Report/Sample Runs/Sample Run 2/Sample_Run_2_Part_2.png
  inflating: EECS 3421 A2/Report/Sample Runs/Sample Run 2/Sample_Run_2_Part_1.png
   creating: EECS 3421 A2/Report/Sample Runs/Sample Run 4/
  inflating: EECS 3421 A2/Report/Sample Runs/Sample Run 4/Sample_Run_4.png
   creating: EECS 3421 A2/Report/Sample Runs/Sample Run 5/
  inflating: EECS 3421 A2/Report/Sample Runs/Sample Run 5/Sample_Run_5.png
   creating: EECS 3421 A2/Report/Sample Runs/Sample Run 6/
  inflating: EECS 3421 A2/Report/Sample Runs/Sample Run 6/Sample_Run_6_Part_1.png
  inflating: EECS 3421 A2/Report/Sample Runs/Sample Run 6/Sample_Run_6_Part_2.png
   creating: EECS 3421 A2/Report/Sample Runs/Sample Run 7/
  inflating: EECS 3421 A2/Report/Sample Runs/Sample Run 7/Sample_Run_7_Part_2.png
  inflating: EECS 3421 A2/Report/Sample Runs/Sample Run 7/Sample_Run_7_Part_1.png
   creating: EECS 3421 A2/Report/Sample Runs/Sample Run 1/
  inflating: EECS 3421 A2/Report/Sample Runs/Sample Run 1/Sample_Run_1_Part_2.png
  inflating: EECS 3421 A2/Report/Sample Runs/Sample Run 1/Sample_Run_1_Part_1.png
   creating: EECS 3421 A2/Report/Sample Runs/Sample Run 3/
  inflating: EECS 3421 A2/Report/Sample Runs/Sample Run 3/Sample_Run_3.png
   creating: EECS 3421 A2/Report/Sample Runs/Sample Run 8/
  inflating: EECS 3421 A2/Report/Sample Runs/Sample Run 8/Sample_Run_8_Part_3.png
  inflating: EECS 3421 A2/Report/Sample Runs/Sample Run 8/Sample_Run_8_Part_2.png
  inflating: EECS 3421 A2/Report/Sample Runs/Sample Run 8/Sample_Run_8_Part_1.png
   creating: EECS 3421 A2/Code/
  inflating: EECS 3421 A2/Code/YRBAPP.java
  inflating: EECS 3421 A2/Code/yrb-create
  inflating: EECS 3421 A2/Code/Purchase.java
  inflating: EECS 3421 A2/Code/YRBAPPUtility.java
  inflating: EECS 3421 A2/Code/run.sh
  inflating: EECS 3421 A2/Code/yrb-drop
  inflating: EECS 3421 A2/Code/Book.java
ashkan $ 
```

3. Allow the script (**run.sh**) to be run as a program (by giving it execution privilege) by either right clicking on the script file and choosing **Properties** to open the run.sh Properties window. At this point navigate to the **Permissions** tab in the properties window and check the box in front of **Execute** which says: **Allow executing file as program** or by entering the following command in the terminal: **chmod +x run.sh**



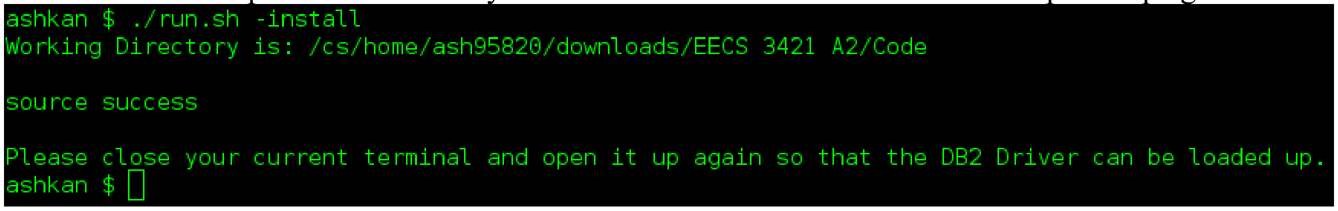
```
ashkan $ chmod +x run.sh
ashkan $ 
```

4. Now please create the YRB database using the following commands in the terminal:
db2 connect to c3421a
db2 -tf yrb-create
db2 connect reset
db2 terminate
or using the script with the **-create** option like in the following: **./run.sh -create** and then compile the application using **javac YRBAPP.java** or using the script with the **-compile** option like the following: **./run.sh -compile**



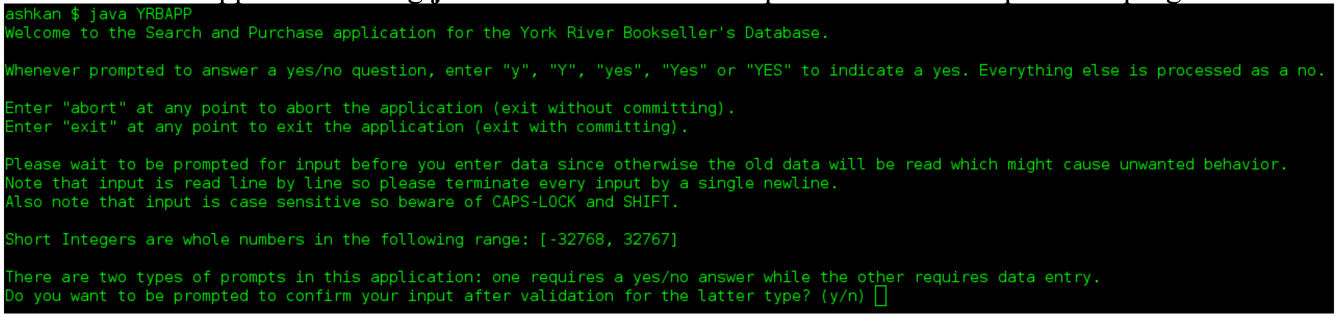
```
ashkan $ ./run.sh -create
Working Directory is: /cs/home/ash95820/downloads/EECS 3421 A2/Code

create success
ashkan $ ./run.sh -compile
Working Directory is: /cs/home/ash95820/downloads/EECS 3421 A2/Code

compile success

Use the following command to run the application: java YRBAPP
ashkan $ 
```

5. Now please install the DB2 driver onto your machine by either entering the following command in the terminal: **source ~db2leduc/cshrc.runtime** or using the script with the **-install** option like the following: **./run.sh -install**. The way the script accomplishes the installation is by appending the sourcing command to **~/.cshrc** only once. Once it has done so, the sourcing will be executed whenever a new terminal is opened which is why it asks the user to close the terminal and open it up again.



```
ashkan $ ./run.sh -install
Working Directory is: /cs/home/ash95820/downloads/EECS 3421 A2/Code

source success

Please close your current terminal and open it up again so that the DB2 Driver can be loaded up.
ashkan $ 
```

6. Now run the application using **java YRBAPP** as the script said after it compiled the program.



```
ashkan $ java YRBAPP
Welcome to the Search and Purchase application for the York River Bookseller's Database.

Whenever prompted to answer a yes/no question, enter "y", "Y", "yes", "Yes" or "YES" to indicate a yes. Everything else is processed as a no.

Enter "abort" at any point to abort the application (exit without committing).
Enter "exit" at any point to exit the application (exit with committing).

Please wait to be prompted for input before you enter data since otherwise the old data will be read which might cause unwanted behavior.
Note that input is read line by line so please terminate every input by a single newline.
Also note that input is case sensitive so beware of CAPS-LOCK and SHIFT.

Short Integers are whole numbers in the following range: [-32768, 32767]

There are two types of prompts in this application: one requires a yes/no answer while the other requires data entry.
Do you want to be prompted to confirm your input after validation for the latter type? (y/n) 
```

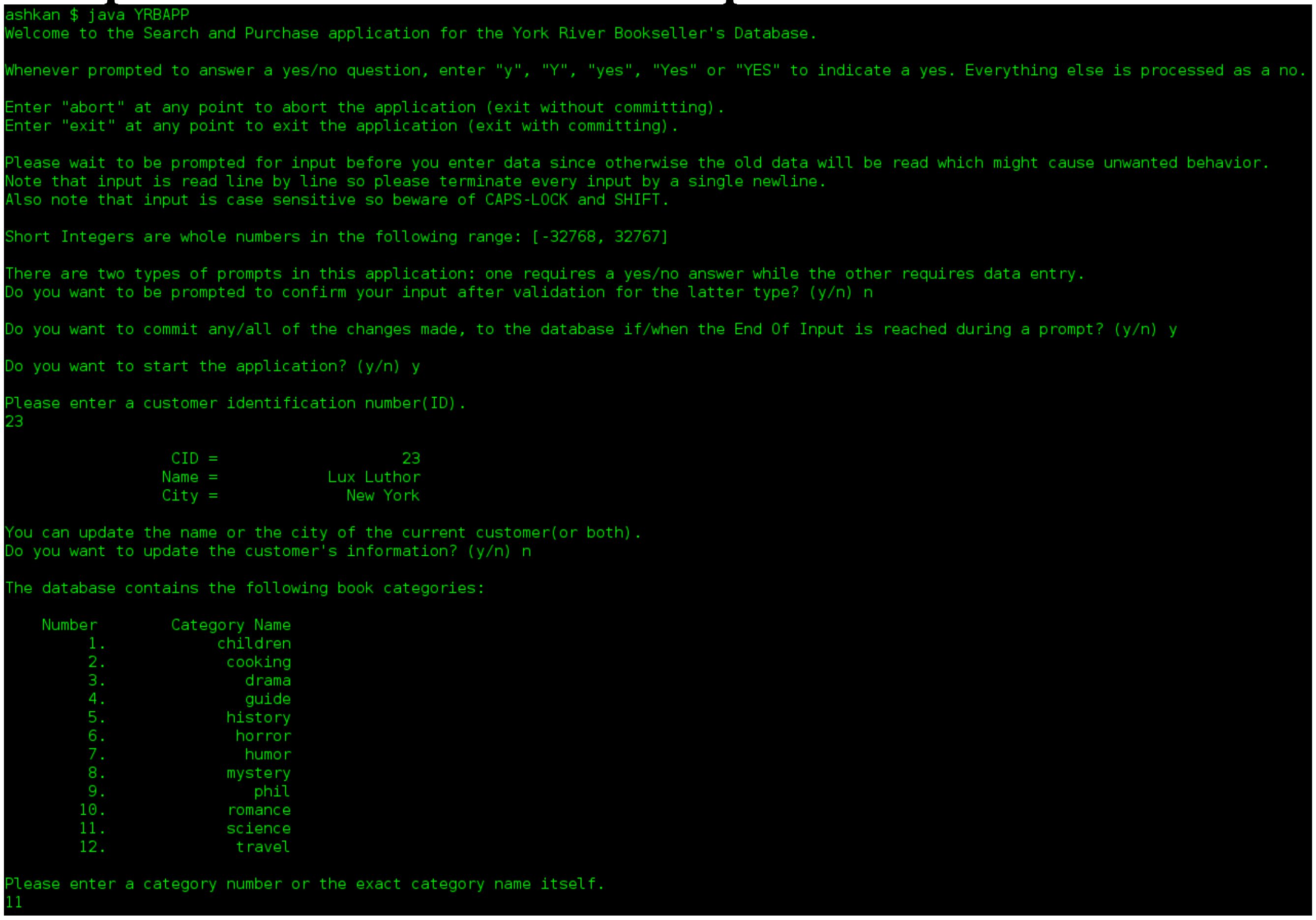
7. The user can now run the application in any way they please to do so by just reading the prompts and answering them in any way they want.

The Program first displays a simple greeting message. It then informs its user of a few simple rules, such as how to answer yes/no questions, what the exiting keywords are and what kind of numbers are valid as input.

At the beginning of every run (including when the application is restarted by the user), it asks if the user wants to be prompted to confirm their data entries which includes the customer ID, the new customer name and city (if an update is requested), the book category and then the book choice and then finally the purchase quantity.

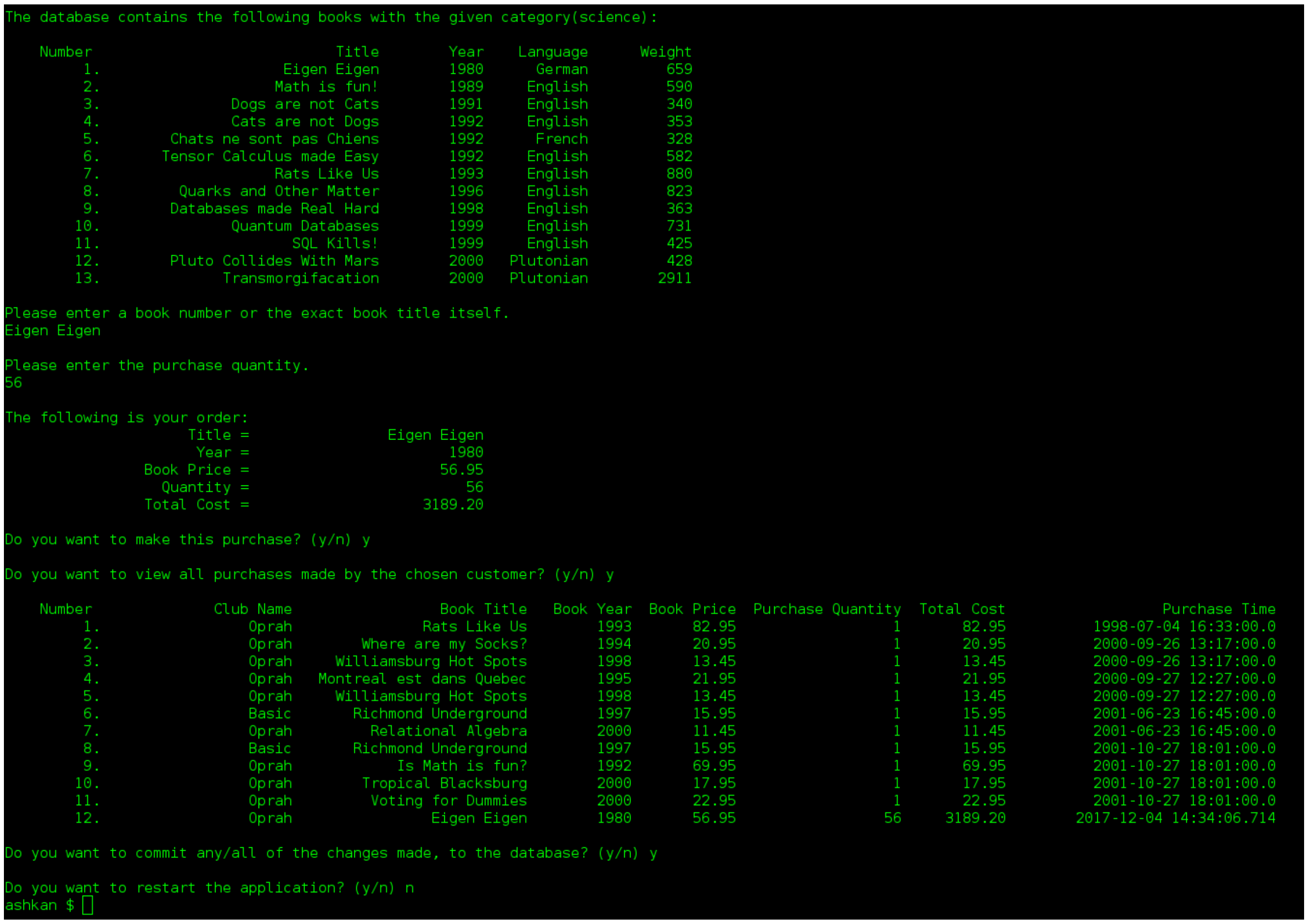
It also prompts for the user to answer a question that is only ever asked once which is what the program should do in case there is no more input (EndOfFile signal has been received).

Sample Run 1: Valid Data/No Update/Purchase



```
ashkan $ java YRBAPP
Welcome to the Search and Purchase application for the York River Bookseller's Database.

Whenever prompted to answer a yes/no question, enter "y", "Y", "yes", "Yes" or "YES" to indicate a yes. Everything else is processed as a no.

Enter "abort" at any point to abort the application (exit without committing).
Enter "exit" at any point to exit the application (exit with committing).

Please wait to be prompted for input before you enter data since otherwise the old data will be read which might cause unwanted behavior.
Note that input is read line by line so please terminate every input by a single newline.
Also note that input is case sensitive so beware of CAPS-LOCK and SHIFT.

Short Integers are whole numbers in the following range: [-32768, 32767]

There are two types of prompts in this application: one requires a yes/no answer while the other requires data entry.
Do you want to be prompted to confirm your input after validation for the latter type? (y/n) n

Do you want to commit any/all of the changes made, to the database if/when the End Of Input is reached during a prompt? (y/n) y

Do you want to start the application? (y/n) y

Please enter a customer identification number(ID).
23

                 CID =                    23
                Name =            Lux Luthor
                City =              New York

You can update the name or the city of the current customer(or both).
Do you want to update the customer's information? (y/n) n

The database contains the following book categories:

    Number        Category Name
         1.            children
         2.             cooking
         3.               drama
         4.               guide
         5.             history
         6.              horror
         7.               humor
         8.             mystery
         9.                phil
        10.             romance
        11.             science
        12.              travel

Please enter a category number or the exact category name itself.
11
```

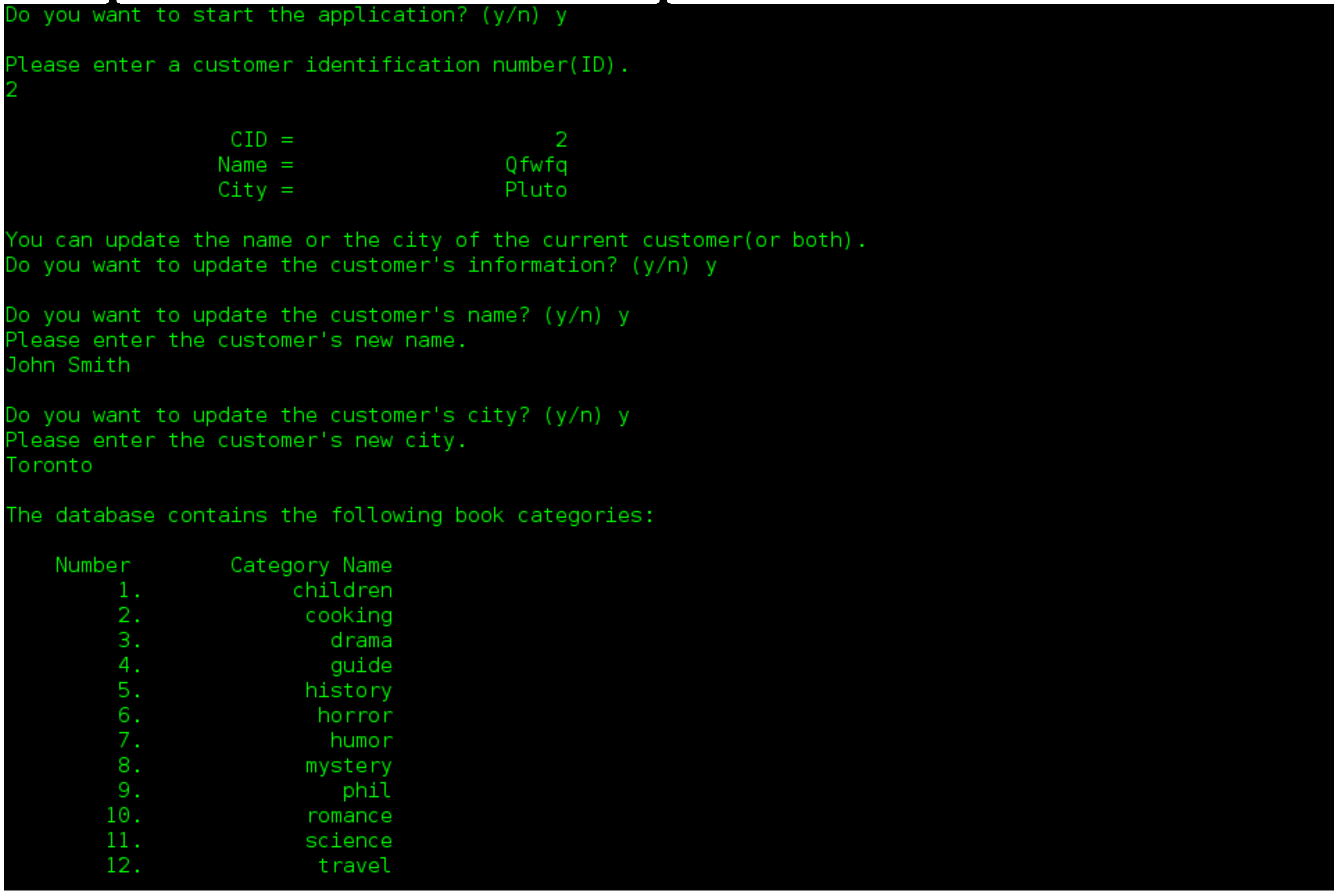
Explanation: In this run, the user chose to not have to confirm their data entry but also to commit the changes if the End Of Input(or End Of File) is reached(Ctrl+D is pressed which sends EOF signal). They then entered a valid short integer(short in Java or smallint in SQL) which also was a valid customer ID(a valid ID is one that exists in the database). After finding the customer, they chose not to update their information and move on to selecting a book category. Instead of typing out the category name (which is also allowed) they chose to enter the number associated to the category they wanted to select.



```
The database contains the following books with the given category(science):

    Number                         Title        Year    Language      Weight
        1.                   Eigen Eigen        1980      German         659
        2.                  Math is fun!        1989     English         590
        3.             Dogs are not Cats        1991     English         340
        4.             Cats are not Dogs        1992     English         353
        5.      Chats ne sont pas Chiens        1992      French         328
        6.     Tensor Calculus made Easy        1992     English         582
        7.                  Rats Like Us        1993     English         880
        8.       Quarks and Other Matter        1996     English         823
        9.      Databases made Real Hard        1998     English         363
       10.             Quantum Databases        1999     English         731
       11.                    SQL Kills!        1999     English         425
       12.      Pluto Collides With Mars        2000   Plutonian         428
       13.            Transmorgifacation        2000   Plutonian        2911

Please enter a book number or the exact book title itself.
Eigen Eigen

Please enter the purchase quantity.
56

The following is your order:
                     Title =                 Eigen Eigen
                      Year =                        1980
                Book Price =                       56.95
                  Quantity =                          56
                Total Cost =                     3189.20

Do you want to make this purchase? (y/n) y

Do you want to view all purchases made by the chosen customer? (y/n) y

    Number              Club Name                  Book Title   Book Year  Book Price  Purchase Quantity  Total Cost                   Purchase Time
        1.                  Oprah                Rats Like Us        1993       82.95                  1       82.95           1998-07-04 16:33:00.0
        2.                  Oprah         Where are my Socks?        1994       20.95                  1       20.95           2000-09-26 13:17:00.0
        3.                  Oprah      Williamsburg Hot Spots        1998       13.45                  1       13.45           2000-09-26 13:17:00.0
        4.                  Oprah    Montreal est dans Quebec        1995       21.95                  1       21.95           2000-09-27 12:27:00.0
        5.                  Oprah      Williamsburg Hot Spots        1998       13.45                  1       13.45           2000-09-27 12:27:00.0
        6.                  Basic        Richmond Underground        1997       15.95                  1       15.95           2001-06-23 16:45:00.0
        7.                  Oprah          Relational Algebra        2000       11.45                  1       11.45           2001-06-23 16:45:00.0
        8.                  Basic        Richmond Underground        1997       15.95                  1       15.95           2001-10-27 18:01:00.0
        9.                  Oprah             Is Math is fun?        1992       69.95                  1       69.95           2001-10-27 18:01:00.0
       10.                  Oprah         Tropical Blacksburg        2000       17.95                  1       17.95           2001-10-27 18:01:00.0
       11.                  Oprah          Voting for Dummies        2000       22.95                  1       22.95           2001-10-27 18:01:00.0
       12.                  Oprah                 Eigen Eigen        1980       56.95                 56     3189.20         2017-12-04 14:34:06.714

Do you want to commit any/all of the changes made, to the database? (y/n) y

Do you want to restart the application? (y/n) n
ashkan $ 
```

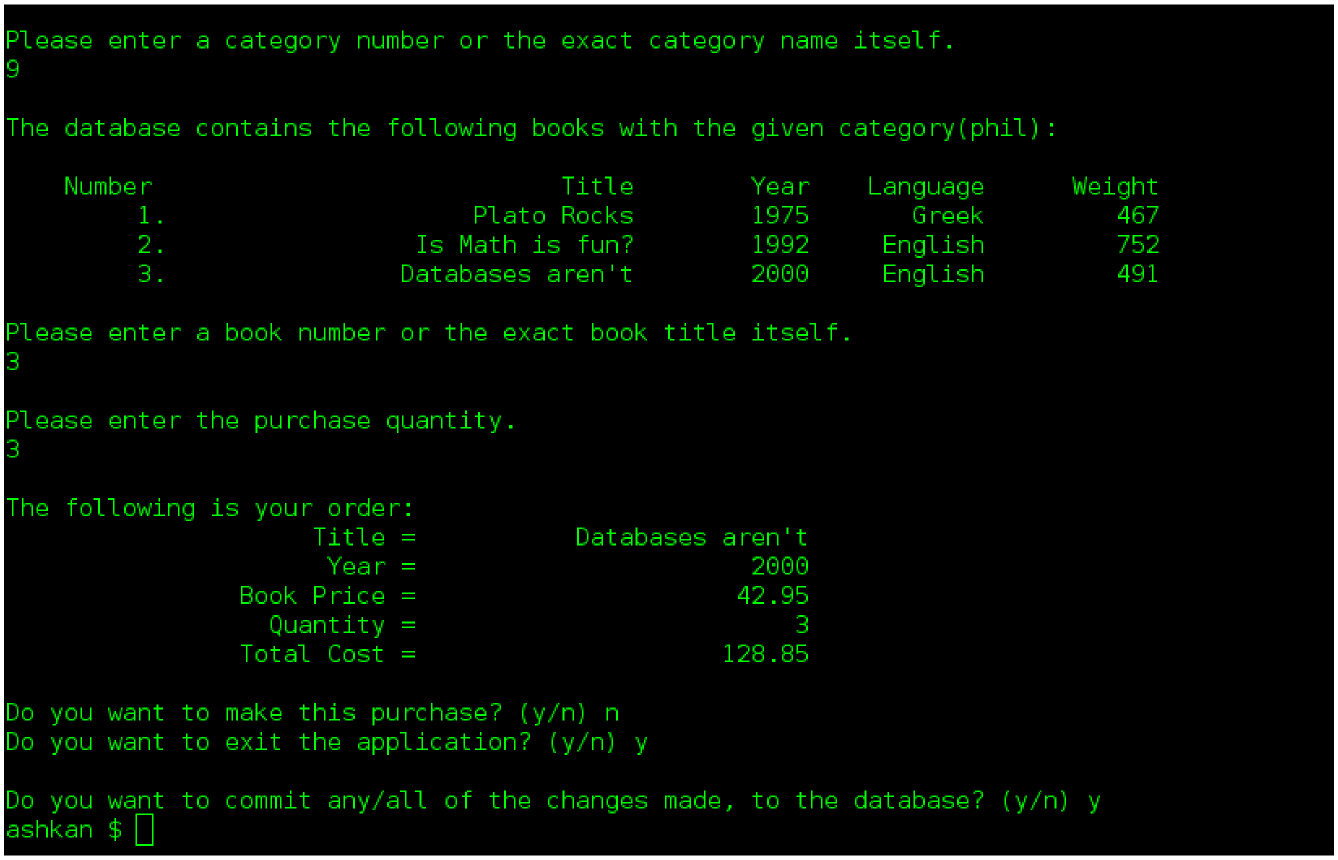
Explanation: After having selected the category, all books of said category were displayed to the user and then they were prompted to select one of the books. This time they chose to actually enter the book title instead of the associated number. The user was then prompted for the purchase quantity. After entering the purchase quantity they chose to make the purchase by saying yes to the appropriate prompt which inserts the new record into the database. They also chose to view all purchases made by this customer that exist in the database. At this point a single run of the application had ended and as such the user was asked if they wanted to commit all of the changes which they chose to do so but did not want to restart the application which is why it just terminated.

Sample Run 2: Valid Data/Update/No Purchase



```
Do you want to start the application? (y/n) y

Please enter a customer identification number(ID).
2

                  CID =                     2
                 Name =                 Qfwfq
                 City =                 Pluto

You can update the name or the city of the current customer(or both).
Do you want to update the customer's information? (y/n) y

Do you want to update the customer's name? (y/n) y
Please enter the customer's new name.
John Smith

Do you want to update the customer's city? (y/n) y
Please enter the customer's new city.
Toronto

The database contains the following book categories:

    Number        Category Name
         1.            children
         2.             cooking
         3.               drama
         4.               guide
         5.             history
         6.              horror
         7.               humor
         8.             mystery
         9.                phil
        10.             romance
        11.             science
        12.              travel
```

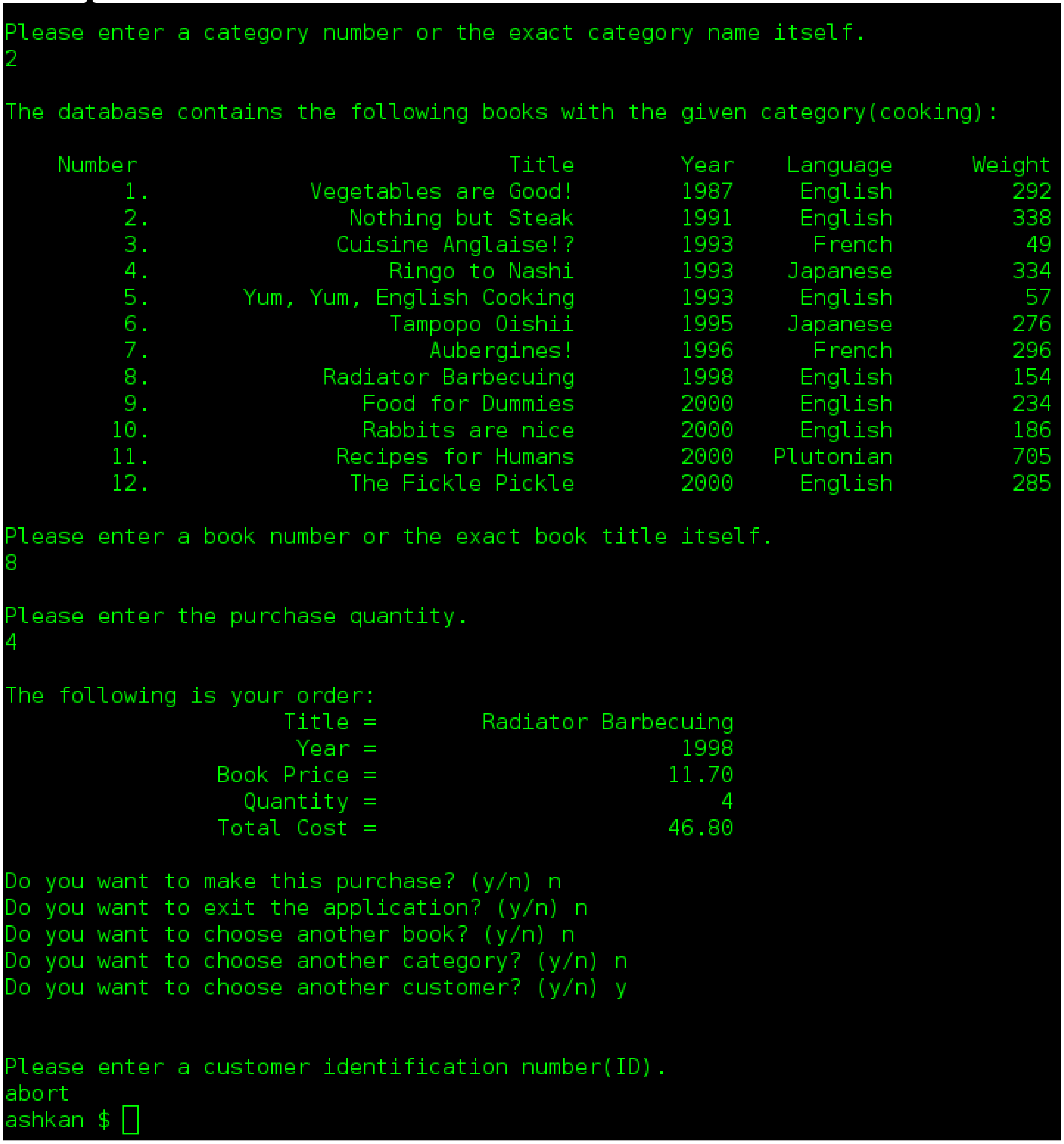
Explanation: In this run, the user acted similarly to **Sample Run 1**. After finding the customer, they chose to update their information before moving on to selecting a book category. The program informs the user that they can only update the customer's name and city. The reason why a customer's ID is not allowed to be updated is that the ID is the primary key and it is also a foreign key in many other tables and as such an update cannot be performed. Instead if one wanted change a customer's ID, they would have to query all information related to them from the database and then delete said information and then reinsert it with the new customer ID. This is however not done by this program which is why it lets its users know exactly what is update-able. It then prompts whether they wish to update the individual attributes of a customer.



```
Please enter a category number or the exact category name itself.
9

The database contains the following books with the given category(phil):

    Number                           Title        Year    Language      Weight
         1.                    Plato Rocks        1975       Greek         467
         2.                Is Math is fun?        1992     English         752
         3.               Databases aren't        2000     English         491

Please enter a book number or the exact book title itself.
3

Please enter the purchase quantity.
3

The following is your order:
                     Title =          Databases aren't
                      Year =                      2000
                Book Price =                     42.95
                  Quantity =                         3
                Total Cost =                    128.85

Do you want to make this purchase? (y/n) n
Do you want to exit the application? (y/n) y

Do you want to commit any/all of the changes made, to the database? (y/n) y
ashkan $
```

Explanation: Similarly to **Sample Run 1** the user entered all of the required purchase information but they chose to not make the purchase by saying no to the appropriate prompt. Since a purchase was not performed, the program asked the user whether they wanted to exit the application which the user agreed to.

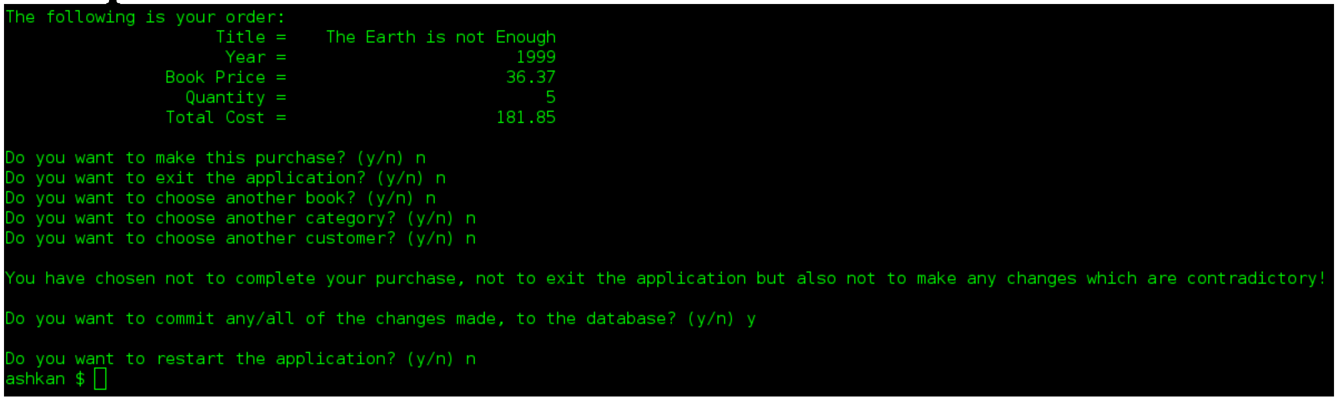
Sample Run 3: Valid Data/No Purchase/Back Track



```
Please enter a category number or the exact category name itself.
2

The database contains the following books with the given category(cooking):

    Number                            Title        Year    Language      Weight
         1.             Vegetables are Good!        1987     English         292
         2.                Nothing but Steak        1991     English         338
         3.               Cuisine Anglaise!?        1993      French          49
         4.                   Ringo to Nashi        1993    Japanese         334
         5.        Yum, Yum, English Cooking        1993     English          57
         6.                   Tampopo Oishii        1995    Japanese         276
         7.                      Aubergines!        1996      French         296
         8.              Radiator Barbecuing        1998     English         154
         9.                  Food for Dummies       2000     English         234
        10.                  Rabbits are nice       2000     English         186
        11.                Recipes for Humans       2000   Plutonian         705
        12.                  The Fickle Pickle      2000     English         285

Please enter a book number or the exact book title itself.
8

Please enter the purchase quantity.
4

The following is your order:
                     Title =        Radiator Barbecuing
                      Year =                       1998
                Book Price =                      11.70
                  Quantity =                          4
                Total Cost =                      46.80

Do you want to make this purchase? (y/n) n
Do you want to exit the application? (y/n) n
Do you want to choose another book? (y/n) n
Do you want to choose another category? (y/n) n
Do you want to choose another customer? (y/n) y


Please enter a customer identification number(ID).
abort
ashkan $ 
```

Explanation: In this run, the user went all the way entering all of the required data but when they got to the purchasing stage, they chose to not make that purchase. The program then went to the back tracking stage where it prompted the user to go back to any of the previous stages and the user agreed to go back to the customer selection stage. However when they were prompted for a customer ID, they entered **abort** which is a keyword for the program that causes it to exit without committing any of the uncommitted changes to the database (it actually performs a rollback).

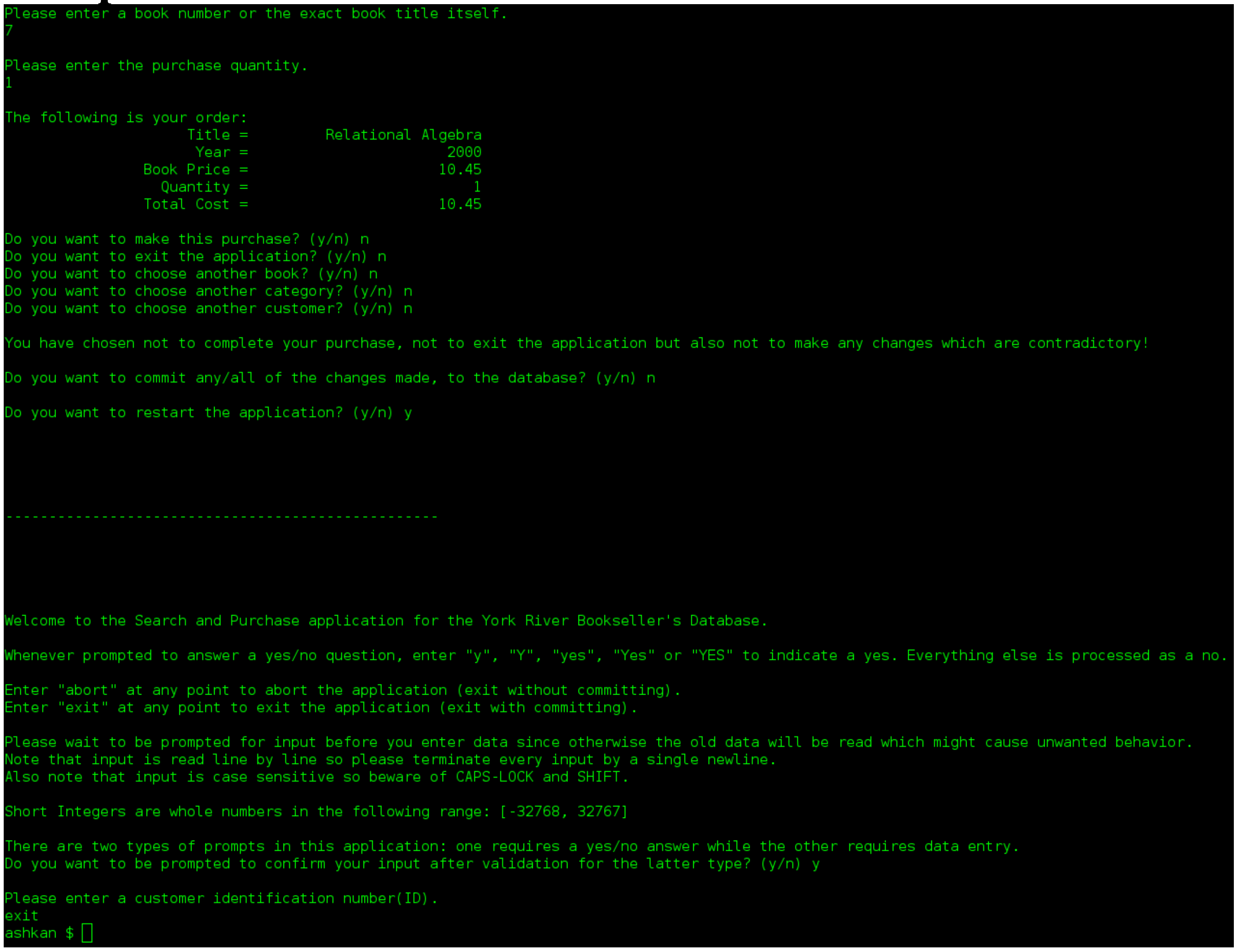
Sample Run 4: Valid Data/No Purchase/No Back Track



```
The following is your order:
                     Title =    The Earth is not Enough
                      Year =                       1999
                Book Price =                      36.37
                  Quantity =                          5
                Total Cost =                     181.85

Do you want to make this purchase? (y/n) n
Do you want to exit the application? (y/n) n
Do you want to choose another book? (y/n) n
Do you want to choose another category? (y/n) n
Do you want to choose another customer? (y/n) n

You have chosen not to complete your purchase, not to exit the application but also not to make any changes which are contradictory!

Do you want to commit any/all of the changes made, to the database? (y/n) y

Do you want to restart the application? (y/n) n
ashkan $ 
```

Explanation: Similar to **Sample Run 3**, the program went to the back tracking stage where it prompted the user to go back to any of the previous stages but the user rejected all of the options. At this point the program informed the user of the odd scenario that it is in and just skipped to ending this run of the application and prompted for committing/rolling back the changes. Afterwards it asked the user if they wished to restart the application which was rejected by the user.

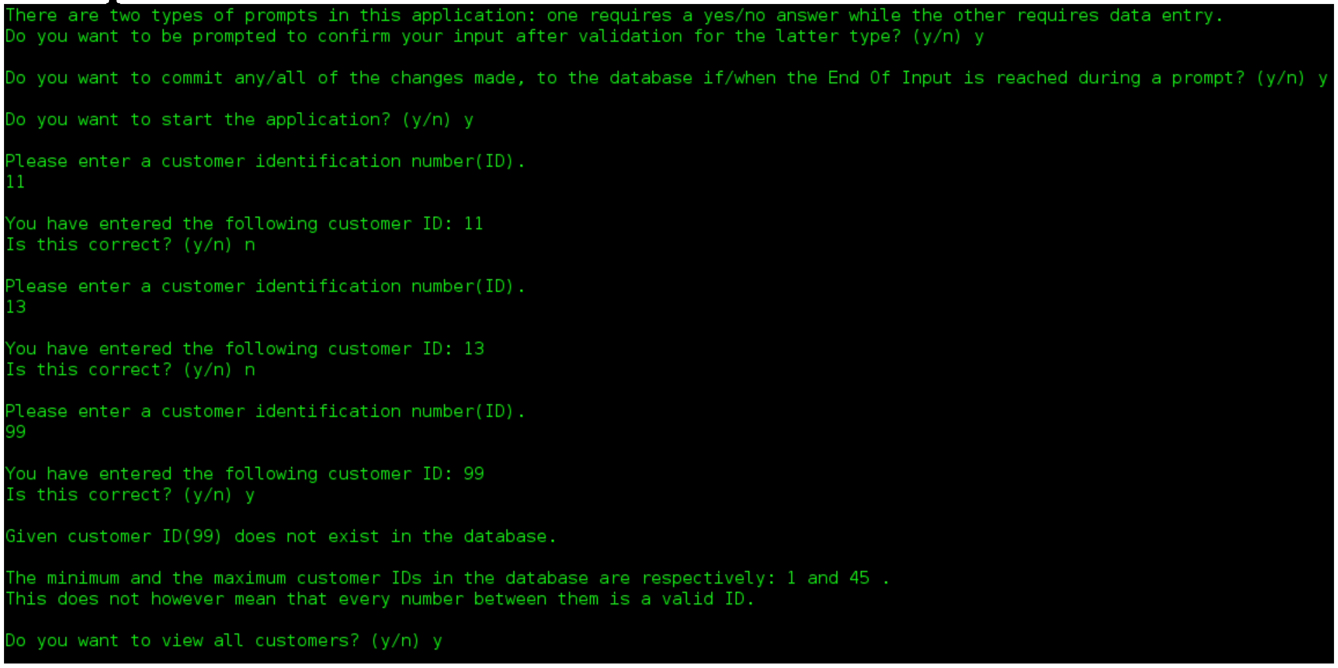
Sample Run 5: Valid Data/No Purchase/Restart



```
Please enter a book number or the exact book title itself.
7

Please enter the purchase quantity.
1

The following is your order:
                     Title =        Relational Algebra
                      Year =                      2000
                Book Price =                     10.45
                  Quantity =                         1
                Total Cost =                     10.45

Do you want to make this purchase? (y/n) n
Do you want to exit the application? (y/n) n
Do you want to choose another book? (y/n) n
Do you want to choose another category? (y/n) n
Do you want to choose another customer? (y/n) n

You have chosen not to complete your purchase, not to exit the application but also not to make any changes which are contradictory!

Do you want to commit any/all of the changes made, to the database? (y/n) n

Do you want to restart the application? (y/n) y





--------------------------------------------------





Welcome to the Search and Purchase application for the York River Bookseller's Database.

Whenever prompted to answer a yes/no question, enter "y", "Y", "yes", "Yes" or "YES" to indicate a yes. Everything else is processed as a no.

Enter "abort" at any point to abort the application (exit without committing).
Enter "exit" at any point to exit the application (exit with committing).

Please wait to be prompted for input before you enter data since otherwise the old data will be read which might cause unwanted behavior.
Note that input is read line by line so please terminate every input by a single newline.
Also note that input is case sensitive so beware of CAPS-LOCK and SHIFT.

Short Integers are whole numbers in the following range: [-32768, 32767]

There are two types of prompts in this application: one requires a yes/no answer while the other requires data entry.
Do you want to be prompted to confirm your input after validation for the latter type? (y/n) y

Please enter a customer identification number(ID).
exit
ashkan $ 
```

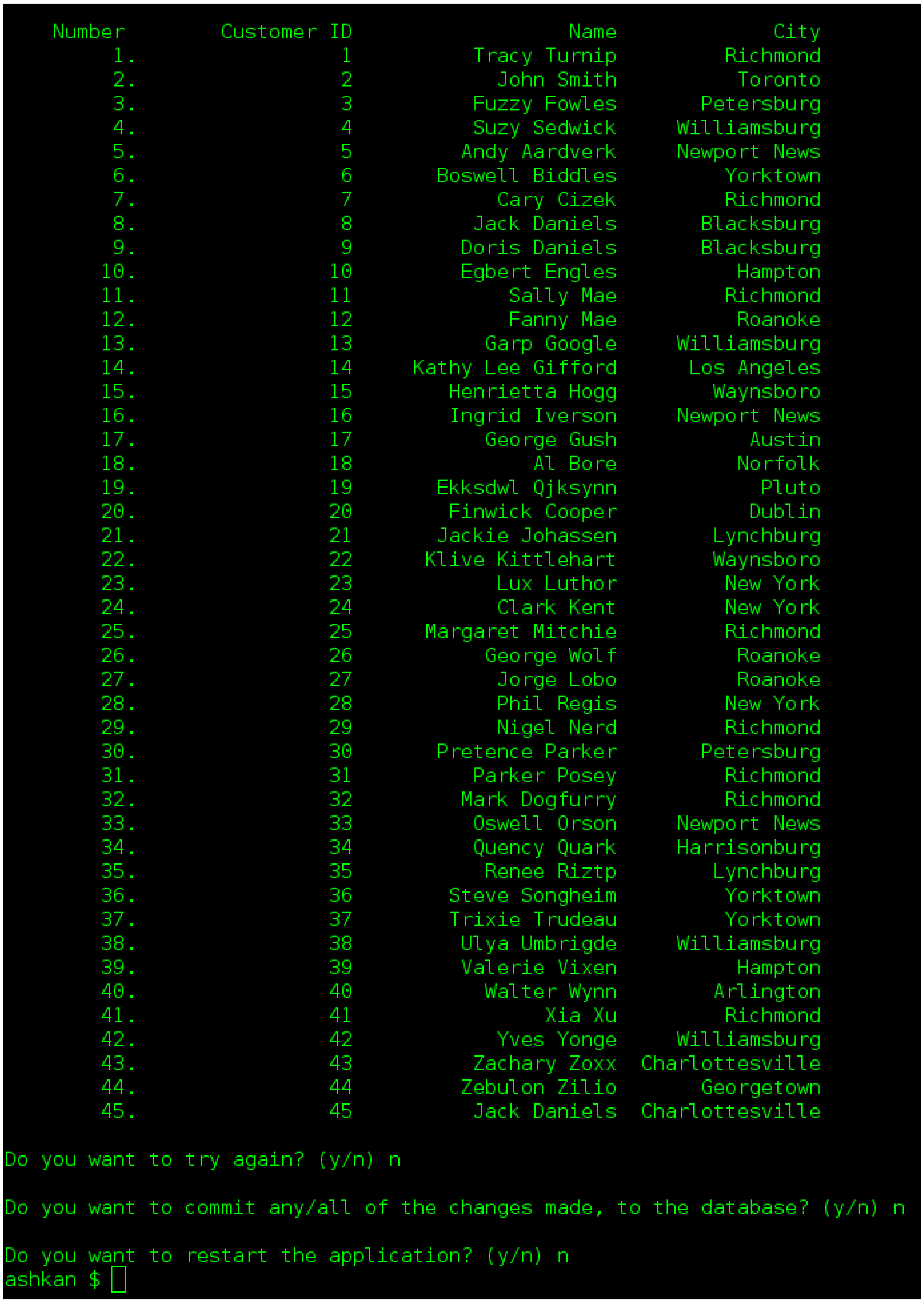
Explanation: Similar to **Sample Run 4**, the program went to the back tracking stage where but the user again rejected all attempts to go back to a previous stage. However they did choose to restart the application entirely which brought them back to the initial messages and prompt. They however chose to enter **exit** which is a keyword for the program that causes it to exit and commit any of the uncommitted changes to the database.

Sample Run 6: Valid Data/Confirm Data



```
There are two types of prompts in this application: one requires a yes/no answer while the other requires data entry.
Do you want to be prompted to confirm your input after validation for the latter type? (y/n) y

Do you want to commit any/all of the changes made, to the database if/when the End Of Input is reached during a prompt? (y/n) y

Do you want to start the application? (y/n) y

Please enter a customer identification number(ID).
11

You have entered the following customer ID: 11
Is this correct? (y/n) n

Please enter a customer identification number(ID).
13

You have entered the following customer ID: 13
Is this correct? (y/n) n

Please enter a customer identification number(ID).
99

You have entered the following customer ID: 99
Is this correct? (y/n) y

Given customer ID(99) does not exist in the database.

The minimum and the maximum customer IDs in the database are respectively: 1 and 45 .
This does not however mean that every number between them is a valid ID.

Do you want to view all customers? (y/n) y
```

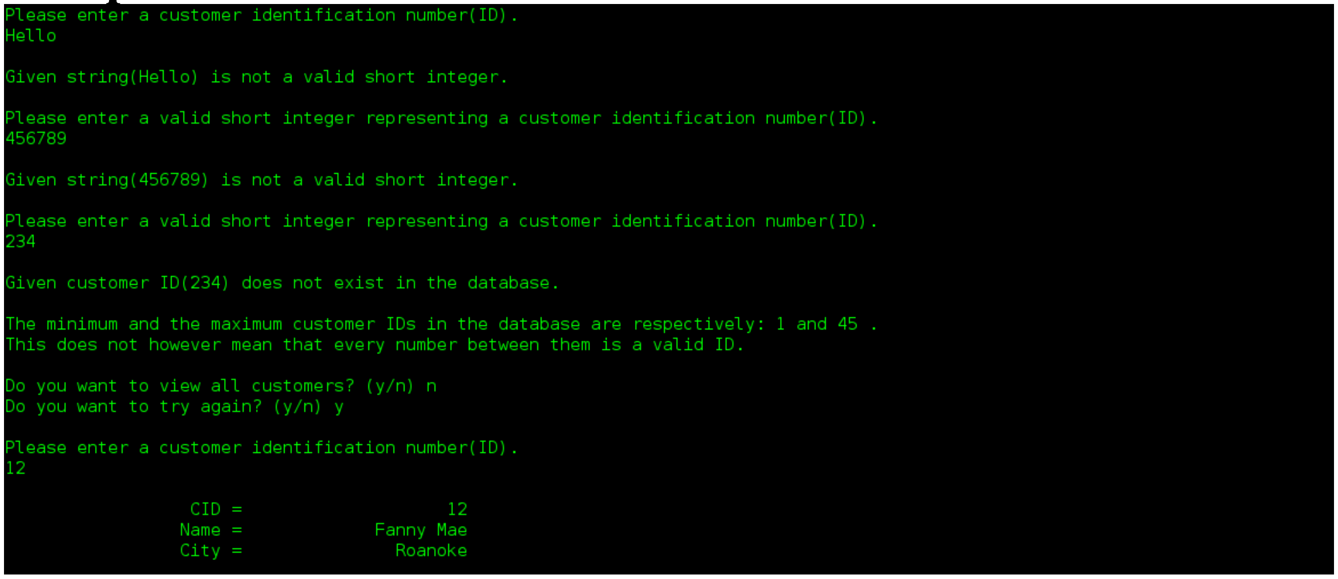
Explanation: In this run, the user actually chose to enable the data confirmation system which allows the user to actually change their choice before moving forward. They entered some valid short integers and then eventually chose to confirm one but unfortunately no customer with that ID existed in the database. The program chose to show them the minimum and maximum customer IDs so that they could get a sense of what the range of IDs is. But as the program says, not every integer in the given range is a valid customer ID which is why it then prompts the user whether they wish to see all customers so that they can potentially make a more informed decision.



```
    Number        Customer ID                Name             City
         1.                 1        Tracy Turnip         Richmond
         2.                 2          John Smith          Toronto
         3.                 3        Fuzzy Fowles       Petersburg
         4.                 4        Suzy Sedwick     Williamsburg
         5.                 5        Andy Aardverk    Newport News
         6.                 6      Boswell Biddles        Yorktown
         7.                 7          Cary Cizek         Richmond
         8.                 8        Jack Daniels       Blacksburg
         9.                 9       Doris Daniels       Blacksburg
        10.                10       Egbert Engles          Hampton
        11.                11           Sally Mae         Richmond
        12.                12           Fanny Mae          Roanoke
        13.                13         Garp Google     Williamsburg
        14.                14   Kathy Lee Gifford      Los Angeles
        15.                15     Henrietta Hogg        Waynsboro
        16.                16     Ingrid Iverson    Newport News
        17.                17         George Gush           Austin
        18.                18             Al Bore          Norfolk
        19.                19     Ekksdwl Qjksynn            Pluto
        20.                20      Finwick Cooper           Dublin
        21.                21     Jackie Johassen        Lynchburg
        22.                22    Klive Kittlehart        Waynsboro
        23.                23          Lux Luthor         New York
        24.                24          Clark Kent         New York
        25.                25    Margaret Mitchie         Richmond
        26.                26         George Wolf          Roanoke
        27.                27          Jorge Lobo          Roanoke
        28.                28          Phil Regis         New York
        29.                29          Nigel Nerd         Richmond
        30.                30     Pretence Parker       Petersburg
        31.                31        Parker Posey         Richmond
        32.                32       Mark Dogfurry         Richmond
        33.                33        Oswell Orson     Newport News
        34.                34       Quency Quark      Harrisonburg
        35.                35         Renee Riztp        Lynchburg
        36.                36      Steve Songheim         Yorktown
        37.                37      Trixie Trudeau         Yorktown
        38.                38       Ulya Umbrigde     Williamsburg
        39.                39      Valerie Vixen           Hampton
        40.                40         Walter Wynn        Arlington
        41.                41              Xia Xu         Richmond
        42.                42          Yves Yonge     Williamsburg
        43.                43        Zachary Zoxx  Charlottesville
        44.                44       Zebulon Zilio       Georgetown
        45.                45        Jack Daniels  Charlottesville

Do you want to try again? (y/n) n

Do you want to commit any/all of the changes made, to the database? (y/n) n

Do you want to restart the application? (y/n) n
ashkan $ 
```

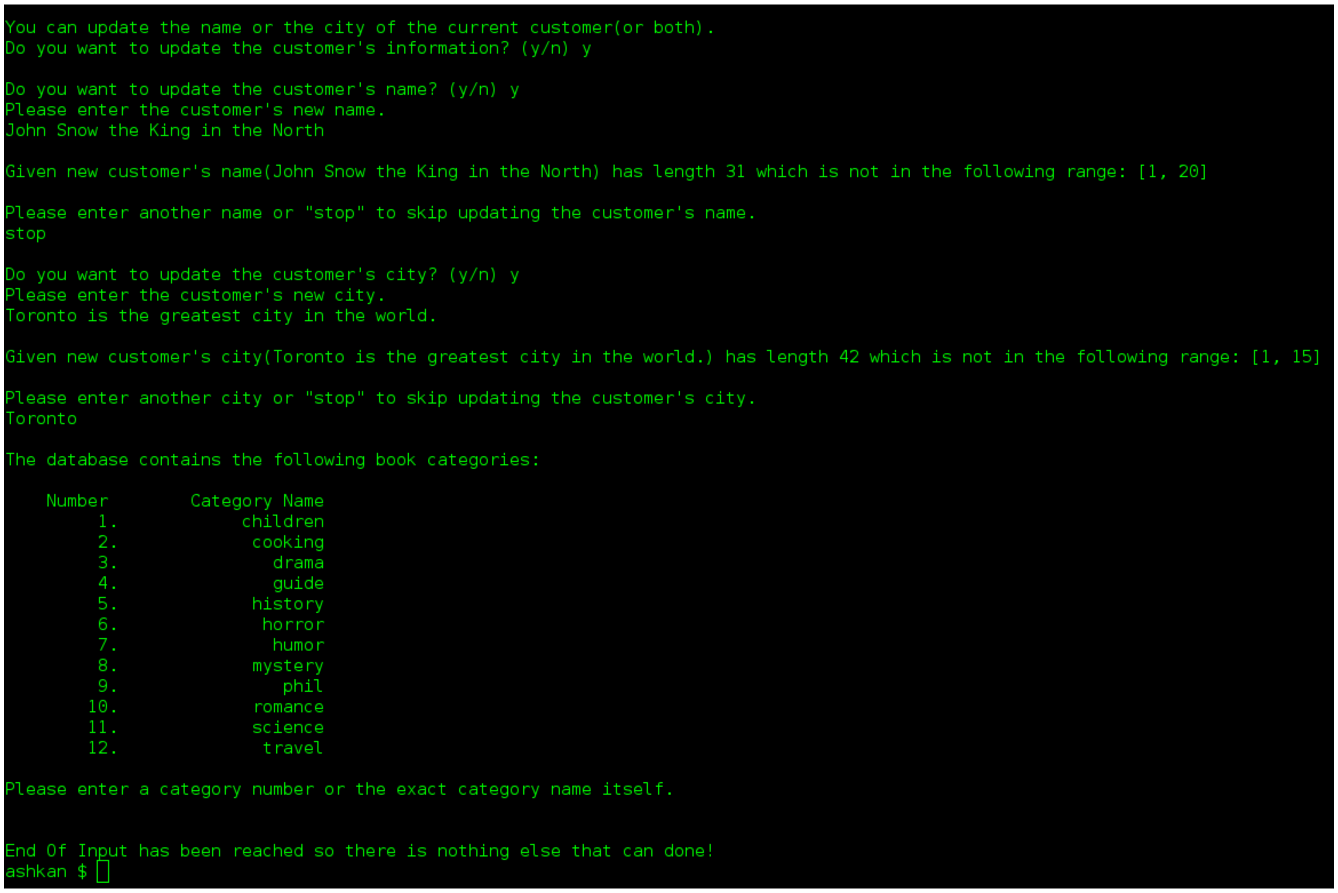
Explanation: Since at this stage a valid customer ID has not yet been read, the program asked the user if they wished to try again and since they refused, it just wanted to the end stage and performed all of the necessary actions in that stage.

Sample Run 7: Invalid Customer Information/EOF



```
Please enter a customer identification number(ID).
Hello

Given string(Hello) is not a valid short integer.

Please enter a valid short integer representing a customer identification number(ID).
456789

Given string(456789) is not a valid short integer.

Please enter a valid short integer representing a customer identification number(ID).
234

Given customer ID(234) does not exist in the database.

The minimum and the maximum customer IDs in the database are respectively: 1 and 45 .
This does not however mean that every number between them is a valid ID.

Do you want to view all customers? (y/n) n
Do you want to try again? (y/n) y

Please enter a customer identification number(ID).
12

                 CID =                    12
                Name =             Fanny Mae
                City =               Roanoke
```

Explanation: Unlike all previous runs, the user attempted to pass invalid data at the customer ID entry stage but every single one was caught and an appropriate message was displayed.



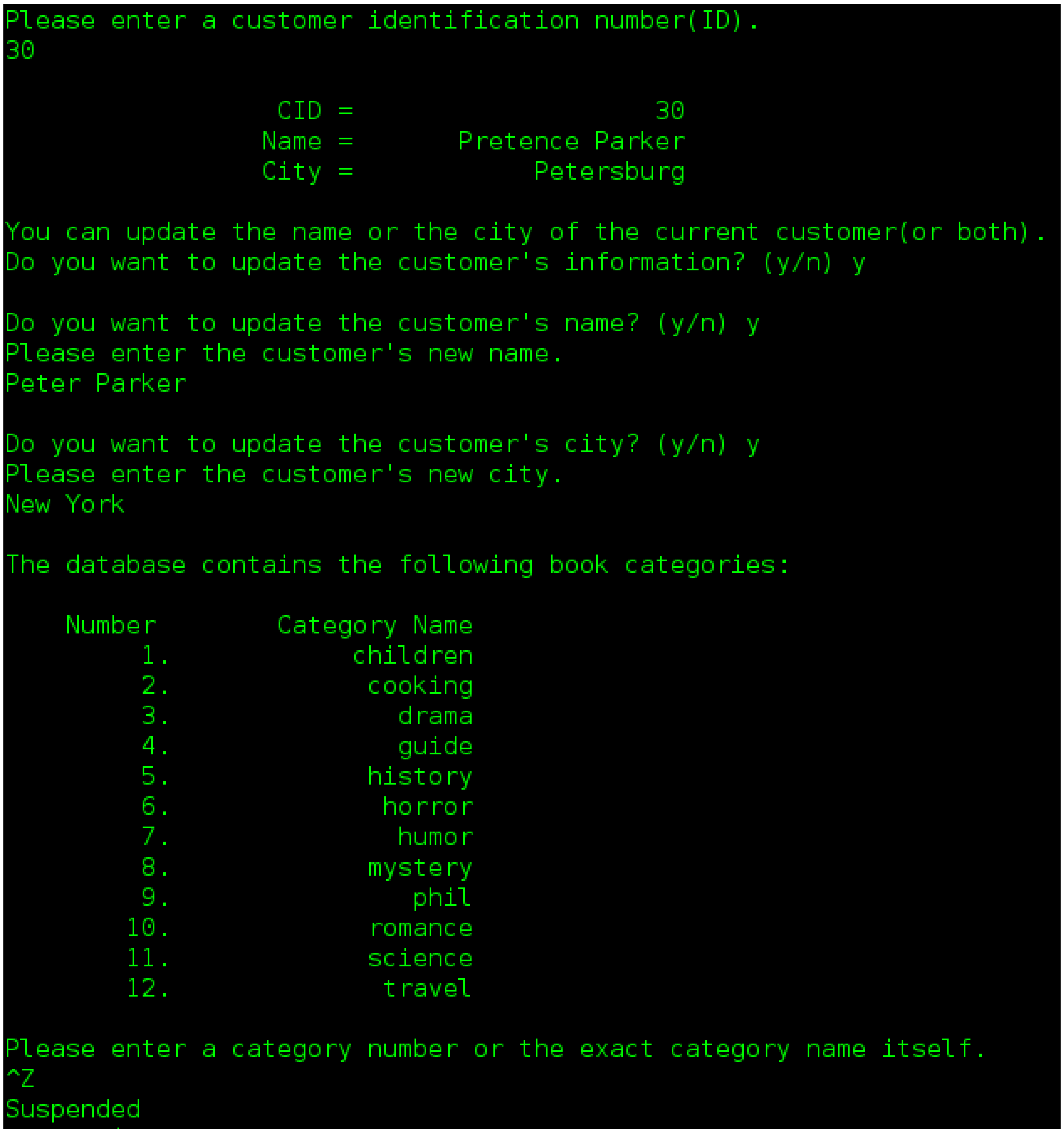
```
You can update the name or the city of the current customer(or both).
Do you want to update the customer's information? (y/n) y

Do you want to update the customer's name? (y/n) y
Please enter the customer's new name.
John Snow the King in the North

Given new customer's name(John Snow the King in the North) has length 31 which is not in the following range: [1, 20]

Please enter another name or "stop" to skip updating the customer's name.
stop

Do you want to update the customer's city? (y/n) y
Please enter the customer's new city.
Toronto is the greatest city in the world.

Given new customer's city(Toronto is the greatest city in the world.) has length 42 which is not in the following range: [1, 15]

Please enter another city or "stop" to skip updating the customer's city.
Toronto

The database contains the following book categories:

    Number        Category Name
         1.            children
         2.             cooking
         3.               drama
         4.               guide
         5.             history
         6.              horror
         7.               humor
         8.             mystery
         9.                phil
        10.             romance
        11.             science
        12.              travel

Please enter a category number or the exact category name itself.


End Of Input has been reached so there is nothing else that can done!
ashkan $ 
```

Explanation: After selecting the customer, the user chose to update the customer's information but they entered strings that were longer than the maximum length specified for the appropriate fields in the database and as such the entry was rejected and they were informed of this action.

Although the database defines the name attribute to be **varchar(20)** which means that it can also be the empty string (a string of length 0) but the program disallows this possibility by rejecting the empty string for a customer name.

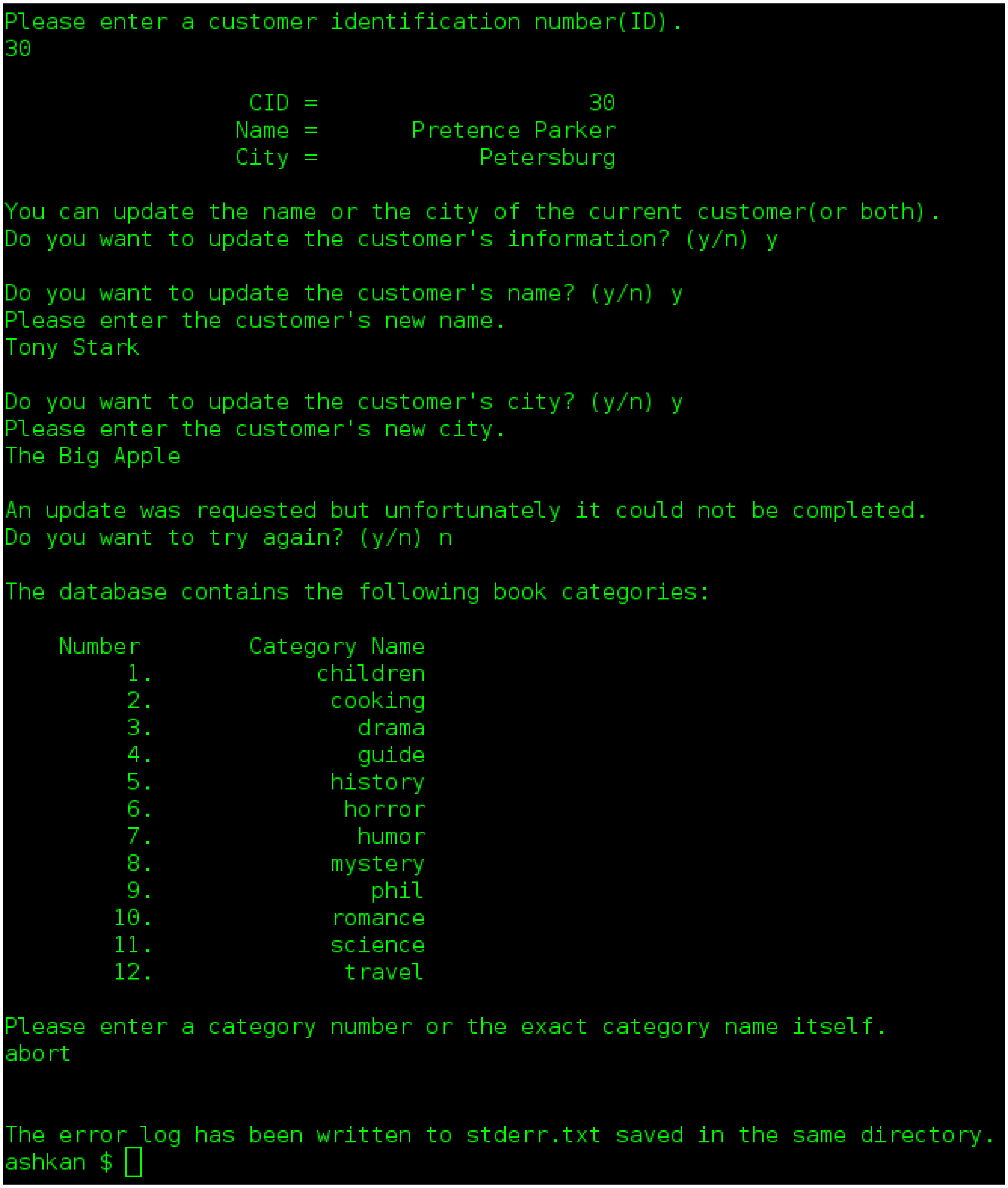
Similarly for a customer city, the empty string is rejected.

Errors Section:



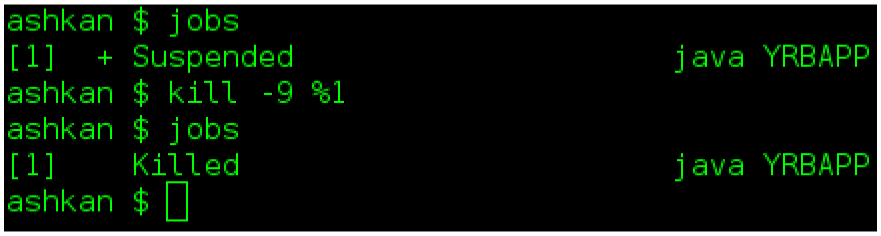
```
Please enter a customer identification number(ID).
30

                 CID =                   30
                Name =       Pretence Parker
                City =            Petersburg

You can update the name or the city of the current customer(or both).
Do you want to update the customer's information? (y/n) y

Do you want to update the customer's name? (y/n) y
Please enter the customer's new name.
Peter Parker

Do you want to update the customer's city? (y/n) y
Please enter the customer's new city.
New York

The database contains the following book categories:

    Number        Category Name
         1.            children
         2.             cooking
         3.               drama
         4.               guide
         5.             history
         6.              horror
         7.               humor
         8.             mystery
         9.                phil
        10.             romance
        11.             science
        12.              travel

Please enter a category number or the exact category name itself.
^Z
Suspended
```

Explanation: In this run, the user chose to update a customer's information but then pressed **Ctrl+Z** after the new data had been entered and the application had executed the update command. Since in this application the auto-committing has been turned off, this version of the application will hold the exclusive locks on that customer tuple in the yrb-customer table.



```
Please enter a customer identification number(ID).
30

                 CID =                   30
                Name =      Pretence Parker
                City =           Petersburg

You can update the name or the city of the current customer(or both).
Do you want to update the customer's information? (y/n) y

Do you want to update the customer's name? (y/n) y
Please enter the customer's new name.
Tony Stark

Do you want to update the customer's city? (y/n) y
Please enter the customer's new city.
The Big Apple

An update was requested but unfortunately it could not be completed.
Do you want to try again? (y/n) n

The database contains the following book categories:

    Number        Category Name
         1.            children
         2.             cooking
         3.               drama
         4.               guide
         5.             history
         6.              horror
         7.               humor
         8.             mystery
         9.                phil
        10.             romance
        11.             science
        12.              travel

Please enter a category number or the exact category name itself.
abort


The error log has been written to stderr.txt saved in the same directory.
ashkan $ 
```

Explanation: Now if the user reruns the application and chooses to update the same customer, there will be an issue since a previous version (the suspended application) is holding the exclusive lock. However since a **10 second** timeout has been set, the currently running application will not hang but instead it will just fail to update and proceeds to prompt the user. Since there was an error (namely the update failure), a log of all errors will be written to a file named **stderr.txt**.



```
ashkan $ jobs
[1]  + Suspended                     java YRBAPP
ashkan $ kill -9 %1
ashkan $ jobs
[1]    Killed                        java YRBAPP
ashkan $ 
```

If the above scenario with the suspended program has been caused, please consult the above screenshot to terminate the suspended application.

As it is demonstrated, you can run the jobs command to see all currently running applications and then you can use **kill -9** with the proper pointer to kill the desired job.

Use **%1** if the desired job has **[1]** or **%2** if it has **[2]** and so on.

The reason **kill -9 (kill -KILL)** is used is to guarantee the job will actually terminate since many of the termination signals may/could be blocked but the **SIGKILL** cannot be which is why it will always terminate the process.

You can learn more about this by entering **man kill** in the terminal and also to see all termination signals, you can enter **kill -l** in the terminal.

Tools Used in the Project:

The **Eclipse Neon** IDE was used to develop the Java code.

Vim was used to write the Bourne Shell Script (**run.sh**).

The following Java libraries were used in the code.
java.sql.Connection;
java.sql.DriverManager;
java.sql.PreparedStatement;
java.sql.ResultSet;
java.sql.SQLException;

java.util.Scanner;

java.util.Map;
java.util.TreeMap;

java.util.concurrent.atomic.AtomicBoolean;
java.util.concurrent.atomic.AtomicInteger;

java.io.FileNotFoundException;
java.io.PrintWriter;
java.io.UnsupportedEncodingException;

The **sql** library classes were used to connect to database and run all of the sql commands which were mostly queries but there was also one update and one insert.

The Scanner class was used to read input from the standard input stream by using **Scanner(System.in)**.

The **Map** classes were used for data storage internally in the program to map numbers to categories or books for user selection.

The **AtomicBoolean** and **AtomicInteger** classes were needed since a **mutable** Boolean or Integer was required but the implementer did not want to add more global variables that are unnecessary.

The **io** library classes were used for writing all of the errors in the run of the program to a file instead of the standard error stream which is directed to standard output stream if it is not redirected by the user. This was done so that the flow of the program would not be disrupted and the user would not immediately see the errors.