CSc 3320: Systems Programming

Spring 2021

Midterm 2: Total points = 100

Assigned: 11th Apr 2021, Sunday 11:59 PM **Submission Deadline: 18th Apr 2021, Sunday, 11.59 PM (No extensions. If your submission is not received by this time then it will NOT be accepted.)**

Submission instructions:

1. Create a Google doc for your submission.
2. Start your responses from page 2 of the document and copy these instructions on page 1.
3. Fill in your name, campus ID and panther # in the fields provided. If this information is missing TWO POINTS WILL BE DEDUCTED.
4. Keep this page 1 intact. If this *submissions instructions* page is missing in your submission TWO POINTS WILL BE DEDUCTED.
5. Start your responses to each QUESTION on a new page.
6. If you are being asked to write code copy the code into a separate txt file and submit that as well. The code should be executable. E.g. if asked for a C script then provide myfile.c so that we can execute that script. In your answer to the specific question, provide the steps on how to execute your file (like a ReadMe).
7. If you are being asked to test code or run specific commands or scripts, provide the evidence of your outputs through a screenshot and/or screen video-recordings and copy the same into the document.
8. Upon completion, download a .PDF version of the google doc document and submit the same along with all the supplementary files (videos, pictures, scripts etc).

Full Name: Rafid H. Shaon

Campus ID: rshaon1

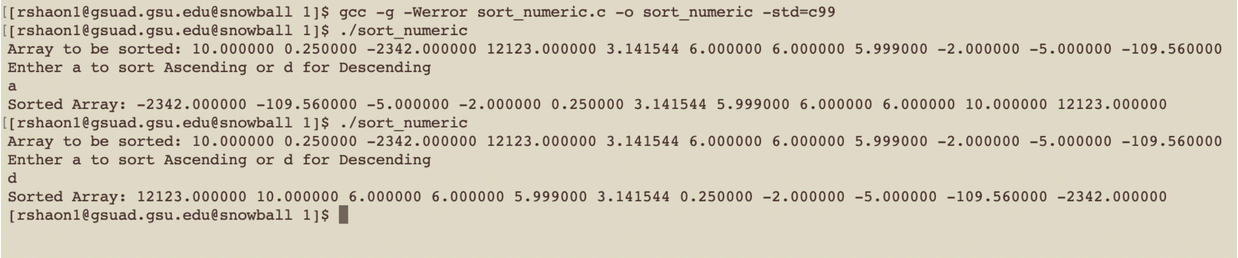
Panther #: 002-49-7367

Questions 1-3 are 20pts each. Question 4 is 40pts
All programs have to be well commented. Non commented programs will receive 0 points. Comments have to be easily comprehensible and concise.

1. Consider the array given below. Write a C program that must be able to sort the elements in the array. You must use pointers in your code to work with the arrays. The sort functionality must be implemented as a separate function named “sort_numeric()”

   Array for your evaluation

   [10, 0.25, -2342, 12123, 3.145435, 6, 6, 5.999, -2, -5, -109.56]
   If given user input A or a: sort in Ascending order
   If given user input D or d: sort in Descending order



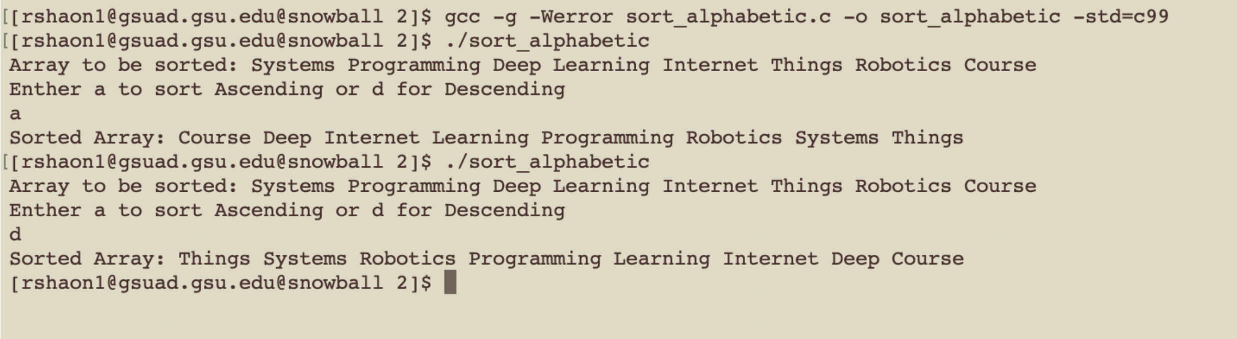
```
[[rshaon1@gsuad.gsu.edu@snowball 1]$ gcc -g -Werror sort_numeric.c -o sort_numeric -std=c99
[[rshaon1@gsuad.gsu.edu@snowball 1]$ ./sort_numeric
 Array to be sorted: 10.000000 0.250000 -2342.000000 12123.000000 3.141544 6.000000 6.000000 5.999000 -2.000000 -5.000000 -109.560000
 Enther a to sort Ascending or d for Descending
 a
 Sorted Array: -2342.000000 -109.560000 -5.000000 -2.000000 0.250000 3.141544 5.999000 6.000000 6.000000 10.000000 12123.000000
[[rshaon1@gsuad.gsu.edu@snowball 1]$ ./sort_numeric
 Array to be sorted: 10.000000 0.250000 -2342.000000 12123.000000 3.141544 6.000000 6.000000 5.999000 -2.000000 -5.000000 -109.560000
 Enther a to sort Ascending or d for Descending
 d
 Sorted Array: 12123.000000 10.000000 6.000000 6.000000 5.999000 3.141544 0.250000 -2.000000 -5.000000 -109.560000 -2342.000000
 [rshaon1@gsuad.gsu.edu@snowball 1]$ █
```

2. Consider the list of names given below. Write a C program that will first create a string array that will contain this list and then sort the elements in the array as per alphabetical order. You must use pointers in your code to work with the arrays. The sort functionality must be implemented as a separate function named “sort_alphabetic()”. The program can be case insensitive (i.e. capital or small letters are treated the same).

List for your evaluation

Systems
Programming
Deep
Learning
Internet
Things
Robotics
Course

If given user input A or a: sort in alphabetical order (a comes first) If
given user input D or d: sort in reverse alphabetical order(z comes first)

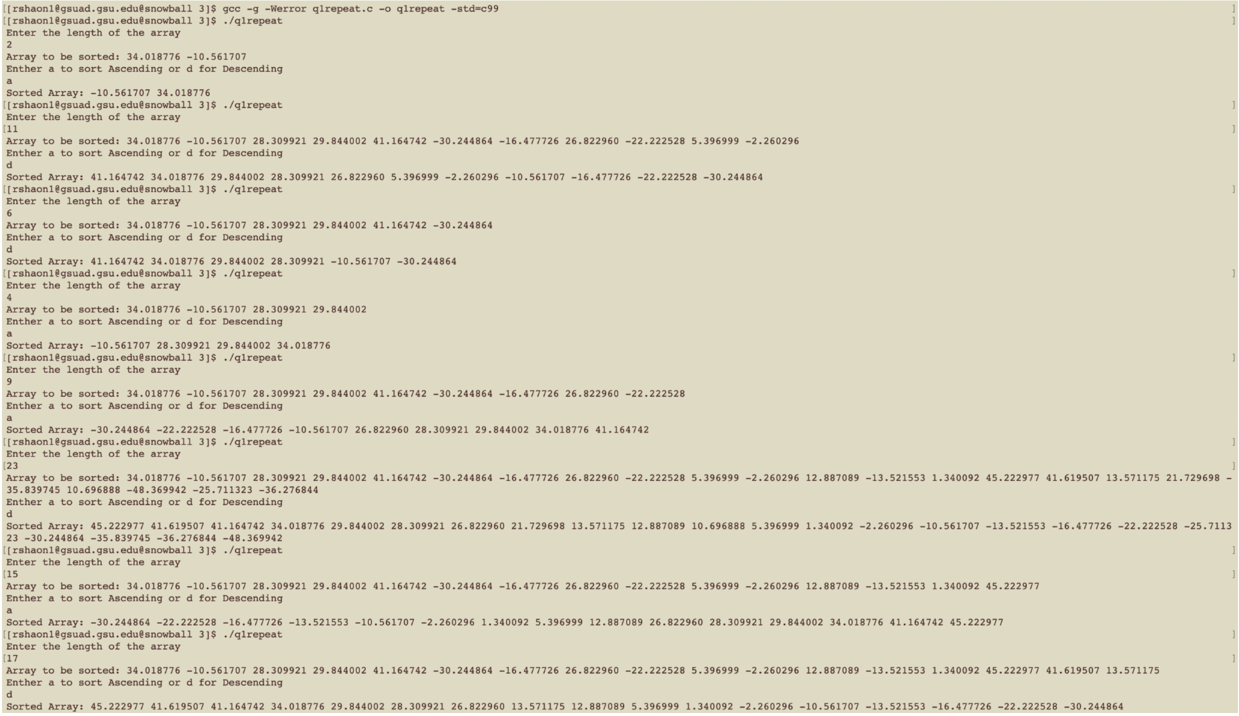


```
[[rshaon1@gsuad.gsu.edu@snowball 2]$ gcc -g -Werror sort_alphabetic.c -o sort_alphabetic -std=c99
[[rshaon1@gsuad.gsu.edu@snowball 2]$ ./sort_alphabetic
 Array to be sorted: Systems Programming Deep Learning Internet Things Robotics Course
 Enther a to sort Ascending or d for Descending
 a
 Sorted Array: Course Deep Internet Learning Programming Robotics Systems Things
[[rshaon1@gsuad.gsu.edu@snowball 2]$ ./sort_alphabetic
 Array to be sorted: Systems Programming Deep Learning Internet Things Robotics Course
 Enther a to sort Ascending or d for Descending
 d
 Sorted Array: Things Systems Robotics Programming Learning Internet Deep Course
 [rshaon1@gsuad.gsu.edu@snowball 2]$ █
```

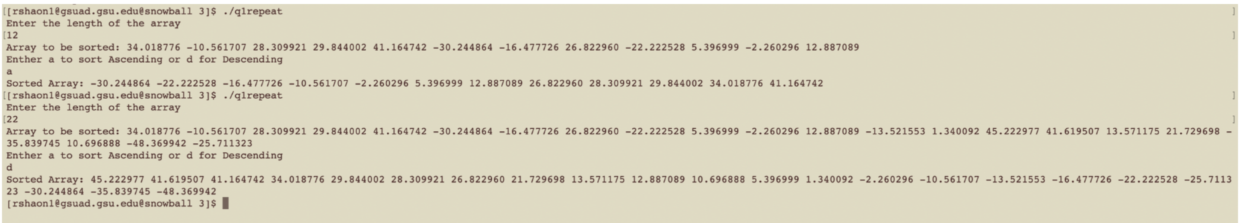
3. Repeat Question 1 or Question 2, considering that the number of elements can potentially increase. That is, the size of the array will be unknown at the start of the program. Note that the requirement of using pointers still holds.

   Show proof of evaluation of your program being able to work for more than 10 entries. Show 5 evaluation trials in your submission. You can pick any number of entries between 10 and 30 for your trials.

   (Hint: *To solve this, use dynamic memory allocation, where you will NOT treat the input array as a known or finite size. Allocate memory space (e.g. malloc()) as and when the number of elements in the list increases).*



```
[[rshaon1@gsuad.gsu.edu@snowball 3]$ gcc -g -Werror q1repeat.c -o q1repeat -std=c99
[[rshaon1@gsuad.gsu.edu@snowball 3]$ ./q1repeat
Enter the length of the array
2
Array to be sorted: 34.018776 -10.561707
Enther a to sort Ascending or d for Descending
a
Sorted Array: -10.561707 34.018776
[[rshaon1@gsuad.gsu.edu@snowball 3]$ ./q1repeat
Enter the length of the array
[11
Array to be sorted: 34.018776 -10.561707 28.309921 29.844002 41.164742 -30.244864 -16.477726 26.822960 -22.222528 5.396999 -2.260296
Enther a to sort Ascending or d for Descending
d
Sorted Array: 41.164742 34.018776 29.844002 28.309921 26.822960 5.396999 -2.260296 -10.561707 -16.477726 -22.222528 -30.244864
[[rshaon1@gsuad.gsu.edu@snowball 3]$ ./q1repeat
Enter the length of the array
6
Array to be sorted: 34.018776 -10.561707 28.309921 29.844002 41.164742 -30.244864
Enther a to sort Ascending or d for Descending
d
Sorted Array: 41.164742 34.018776 29.844002 28.309921 -10.561707 -30.244864
[[rshaon1@gsuad.gsu.edu@snowball 3]$ ./q1repeat
Enter the length of the array
4
Array to be sorted: 34.018776 -10.561707 28.309921 29.844002
Enther a to sort Ascending or d for Descending
a
Sorted Array: -10.561707 28.309921 29.844002 34.018776
[[rshaon1@gsuad.gsu.edu@snowball 3]$ ./q1repeat
Enter the length of the array
9
Array to be sorted: 34.018776 -10.561707 28.309921 29.844002 41.164742 -30.244864 -16.477726 26.822960 -22.222528
Enther a to sort Ascending or d for Descending
a
Sorted Array: -30.244864 -22.222528 -16.477726 -10.561707 26.822960 28.309921 29.844002 34.018776 41.164742
[[rshaon1@gsuad.gsu.edu@snowball 3]$ ./q1repeat
Enter the length of the array
[23
Array to be sorted: 34.018776 -10.561707 28.309921 29.844002 41.164742 -30.244864 -16.477726 26.822960 -22.222528 5.396999 -2.260296 12.887089 -13.521553 1.340092 45.222977 41.619507 13.571175 21.729698 -
35.839745 10.696888 -48.369942 -25.711323 -36.276844
Enther a to sort Ascending or d for Descending
d
Sorted Array: 45.222977 41.619507 41.164742 34.018776 29.844002 28.309921 26.822960 21.729698 13.571175 12.887089 10.696888 5.396999 1.340092 -2.260296 -10.561707 -13.521553 -16.477726 -22.222528 -25.7113
23 -30.244864 -35.839745 -36.276844 -48.369942
[[rshaon1@gsuad.gsu.edu@snowball 3]$ ./q1repeat
Enter the length of the array
[15
Array to be sorted: 34.018776 -10.561707 28.309921 29.844002 41.164742 -30.244864 -16.477726 26.822960 -22.222528 5.396999 -2.260296 12.887089 -13.521553 1.340092 45.222977
Enther a to sort Ascending or d for Descending
a
Sorted Array: -30.244864 -22.222528 -16.477726 -13.521553 -10.561707 -2.260296 1.340092 5.396999 12.887089 26.822960 28.309921 29.844002 34.018776 41.164742 45.222977
[[rshaon1@gsuad.gsu.edu@snowball 3]$ ./q1repeat
Enter the length of the array
[17
Array to be sorted: 34.018776 -10.561707 28.309921 29.844002 41.164742 -30.244864 -16.477726 26.822960 -22.222528 5.396999 -2.260296 12.887089 -13.521553 1.340092 45.222977 41.619507 13.571175
Enther a to sort Ascending or d for Descending
d
Sorted Array: 45.222977 41.619507 41.164742 34.018776 29.844002 28.309921 26.822960 13.571175 12.887089 5.396999 1.340092 -2.260296 -10.561707 -13.521553 -16.477726 -22.222528 -30.244864
```



```
[[rshaon1@gsuad.gsu.edu@snowball 3]$ ./q1repeat
 Enter the length of the array
[12
 Array to be sorted: 34.018776 -10.561707 28.309921 29.844002 41.164742 -30.244864 -16.477726 26.822960 -22.222528 5.396999 -2.260296 12.887089
 Enther a to sort Ascending or d for Descending
 a
 Sorted Array: -30.244864 -22.222528 -16.477726 -10.561707 -2.260296 5.396999 12.887089 26.822960 28.309921 29.844002 34.018776 41.164742
[[rshaon1@gsuad.gsu.edu@snowball 3]$ ./q1repeat
 Enter the length of the array
[22
 Array to be sorted: 34.018776 -10.561707 28.309921 29.844002 41.164742 -30.244864 -16.477726 26.822960 -22.222528 5.396999 -2.260296 12.887089 -13.521553 1.340092 45.222977 41.619507 13.571175 21.729698 -
 35.839745 10.696888 -48.369942 -25.711323
 Enther a to sort Ascending or d for Descending
 d
 Sorted Array: 45.222977 41.619507 41.164742 34.018776 29.844002 28.309921 26.822960 21.729698 13.571175 12.887089 10.696888 5.396999 1.340092 -2.260296 -10.561707 -13.521553 -16.477726 -22.222528 -25.7113
 23 -30.244864 -35.839745 -48.369942
 [rshaon1@gsuad.gsu.edu@snowball 3]$ █
```

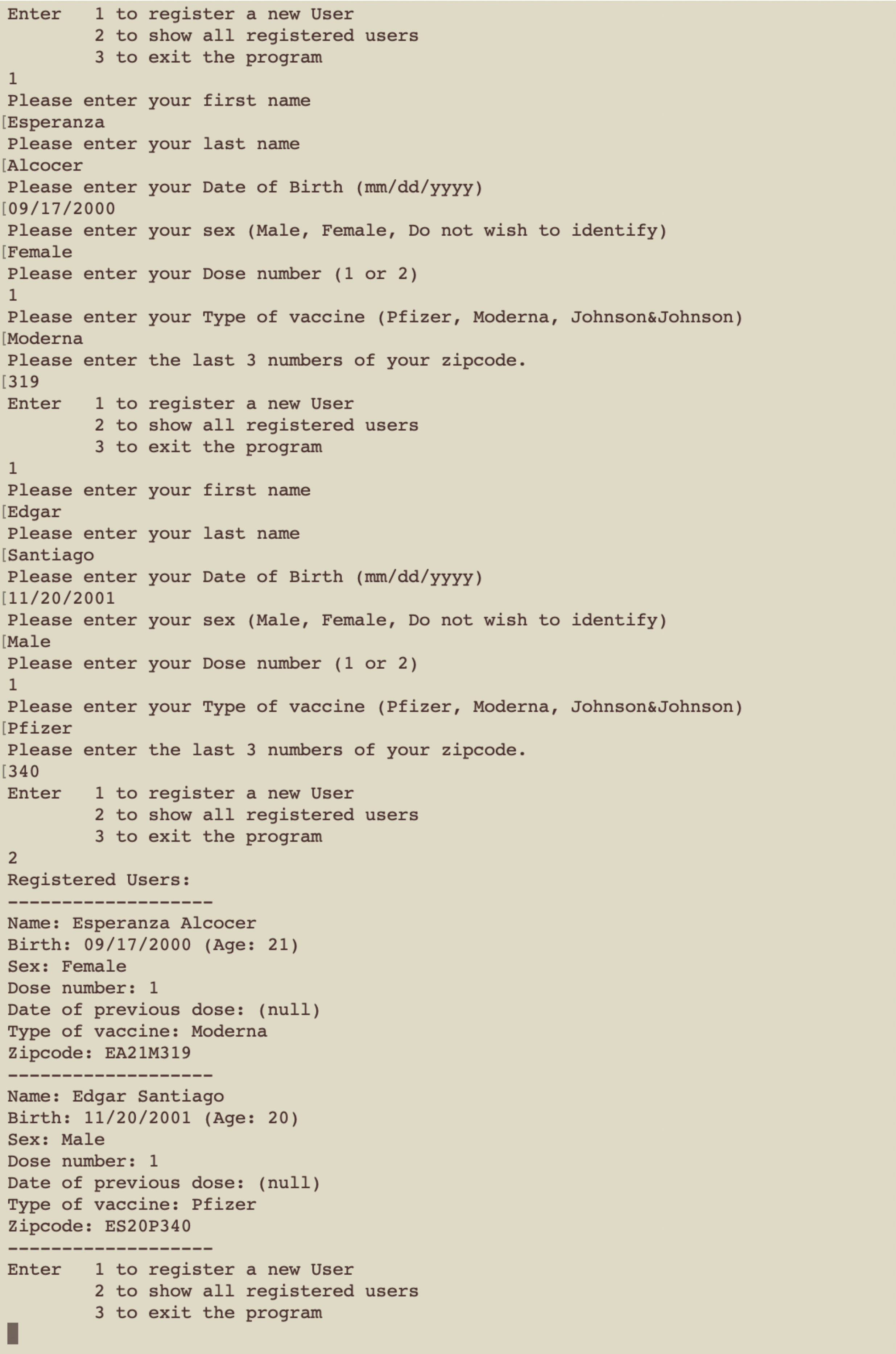
4. Using C programming and using Structures or Unions in your program, build a COVID vaccine registration form where any user can register by filling in their First Name, Last Name, Date of Birth (mm/dd/yyyy), Sex, Dose number (1 or 2), Date of previous dose, Type of vaccine (Pfizer, Moderna, Johnson&Johnson), Residential zipcode.

Upon registration, the system must output a 8 letter alphanumeric code that will be unique to that user. The code is generated as <First letter of First Name><First Letter of Last Name><current age of user -as of registration date><First letter of Vaccine type><last 3 numbers of zipcode>

Add functionality in your program such that it will display all the user's information on the screen (one item in each line).

Show an evaluation trial for registering at least 10 users. For registration, ,for relevant questions, users must choose values based on the options provided (e.g. sex; options must be Male/Female/Do not wish to identify)

(Hint: Write a program that contains main(), register(), generate_code() and retrieve() functions, at the least).



```
 Enter   1 to register a new User
         2 to show all registered users
         3 to exit the program
 1
 Please enter your first name
[Esperanza
 Please enter your last name
[Alcocer
 Please enter your Date of Birth (mm/dd/yyyy)
[09/17/2000
 Please enter your sex (Male, Female, Do not wish to identify)
[Female
 Please enter your Dose number (1 or 2)
 1
 Please enter your Type of vaccine (Pfizer, Moderna, Johnson&Johnson)
[Moderna
 Please enter the last 3 numbers of your zipcode.
[319
 Enter   1 to register a new User
         2 to show all registered users
         3 to exit the program
 1
 Please enter your first name
[Edgar
 Please enter your last name
[Santiago
 Please enter your Date of Birth (mm/dd/yyyy)
[11/20/2001
 Please enter your sex (Male, Female, Do not wish to identify)
[Male
 Please enter your Dose number (1 or 2)
 1
 Please enter your Type of vaccine (Pfizer, Moderna, Johnson&Johnson)
[Pfizer
 Please enter the last 3 numbers of your zipcode.
[340
 Enter   1 to register a new User
         2 to show all registered users
         3 to exit the program
 2
 Registered Users:
 -------------------
 Name: Esperanza Alcocer
 Birth: 09/17/2000 (Age: 21)
 Sex: Female
 Dose number: 1
 Date of previous dose: (null)
 Type of vaccine: Moderna
 Zipcode: EA21M319
 -------------------
 Name: Edgar Santiago
 Birth: 11/20/2001 (Age: 20)
 Sex: Male
 Dose number: 1
 Date of previous dose: (null)
 Type of vaccine: Pfizer
 Zipcode: ES20P340
 -------------------
 Enter   1 to register a new User
         2 to show all registered users
         3 to exit the program
```