PROGRAMMING ASSIGNMENT 5

Subject: Recursion, File IO and Plotting
TAs: Cemil ZALLUHOGLU, Necva BOLUCU, Selma DILEK, Selim YILMAZ

Due Date: 27.12.2017 23:59
Accept your *5th Assignment*.

Problem1: Diamond-Printing Program in C.

Write a program that prints the following diamond shape corresponding to the input value in C Programming language. You must get input number from command line.

input: 4

output :

```
   /\
  /  \
 /    \
/      \
\      /
 \    /
  \  /
   \/
```

Problem2: DVD Store Automation

In this experiment you are expected to develop a simple console based DVD Store Automation. Your application should support the following features:

Load: When your program starts, it should look for a file of which name was taken as a command line argument. If it is found, all records should be read in according to the format explained below. If it cannot be found, your program should initialize an empty inventory. In either case, you will fall into a loop waiting for console commands.

Save data: This operation should be carried out automatically just before your program terminates. All data in your inventory should be saved to the same file (If you run your program several times, each run should input the previous output, without an error.)

User commands:

A - Adding a new entry (Add a new dvd to your inventory)
First get the serial number from the user. If this number is already in use, give a warning and repeat this process. Then get the rest of the data from the user. The state of a new dvd is Inv (in inventory)

```
Input Definition: A,{serial},{price},{Name},{Genre},{Director},{State}
```

```
input: A,123,25,"Terminator","Action","James Cameron","Inv"
```

R - Removing an entry
To remove an item, first carry out the searching by serial in order to see the details of searched dvd. First, user will enter the serial of the dvd and then attributes of the entry will be printed. After that, program asks user to confirm the deletion. If the items state is hired, you cannot remove the item and you must display a warning message.

```
Input Definition: R,{serial}
```

```
input: R,123
```

S - Searching for an entry by name
A search can be conducted either by the name data (multiple items may be listed). If an entry is found, you have to output all the details (name, director, serial etc) . Your code must get the entries in partial search(min. 3 character). (For Ex: If the entry is Ter , your program will get Terminator)

```
Input Definition: S,{Name}
```

```
input: S,"Ter"
input: S,"Terminator"
```

L List
List all the items in the inventory. List each item on one line (print a header row at the top). Print items in groups of 4. If there are more items, wait until the space key is hit before proceeding to the next batch of items. List will be ordered by name.

```
Input Definition: L
```

```
input: L
```

E - Edit entry
Change the values of a record. You have to write on a single line. It is enough to write which attribute you want to edit.

```
Input Definition: E,{Serial},Attribute1=NewValue1,Attribute2=NewValue2,...
```

```
input: E,123,{Name="IronMan"},{Price=50}
```

R- Hire DVD

Change the state of the dvd. First ask user for the serial of the dvd and print its attributes then change its state as Hired. A hired dvd cannot be removed.

```
Input Definition: H,{Serial}
```

```
input: H,123
```

Q- Quit

Save all data into file and end program.

```
Input Definition: Q
```

```
input: Q
```

User will choose an operation from a menu.

A sample menu shown below:

 Note: User can enter A, R, S, L, E, H or Q. If user enters any other character, your application must display an error message.

Some details:

 The inventory will be filled using data in the file of which format is given below. Note that text fields in the file may include whitespace (space or tab) characters. It may be wise to scan all lines for consecutive whitespace characters before loading data into your inventory. Just remove extra whitespaces, leaving only one space character. There can be empty lines in the input file.

 Input file format (each input line consists of the following fields):

 Serial (This is a unique integer. An item can have just one serial and a serial can be used for only one item.).

 Price (Price of the dvd, may have leading blanks)

 Name (Name can include whitespace and punctuation characters)

 Genre (Type can include whitespace characters) Genre of a movie can be "Action", "Fantastic", "Love", "History" and "Horror".

 Director (Director field can include whitespace character)

State: It can be "Hired" or "Inv".

You **must check and give appropriate** warning messages (item couldnt be found, serial number already exists, etc.) whenever necessary.

```
Sample Dvd information

serial: 123
price: 25
director: Terminator
genre: Action
name:   James Cameron
state: Hired
```

Sample Output File

Serial	Price	Name	Genre	Director	State
123	25	Terminator	Action	James Cameron	Hired
456	15	Return of the Jedi	Fantastic	George Lucas	Inv

Problem3: Searching a value in File

In this experiment you are required to implement Binary Search code by using recursion methods.

Binary Search: In computer science, binary search is a search algorithm that finds the position of a target value within a sorted array. Binary search compares the target value to the middle element of the array; if they are unequal, the half in which the target cannot lie is eliminated and the search continues on the remaining half until it is successful. If the search ends with the remaining half being empty, the target is not in the array. `https://en.wikipedia.org/wiki/Binary_search_algorithm`

Here we have an example of binary search algorithm.
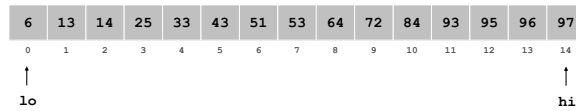
Ex. Binary search for 33.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

↑ lo   ↑ hi

Figure 1: Step1

Ex. Binary search for 33.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

↑ lo   ↑ mid   ↑ hi
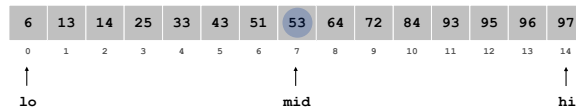
Figure 2: Step2

Ex. Binary search for 33.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

↑ lo   ↑ hi
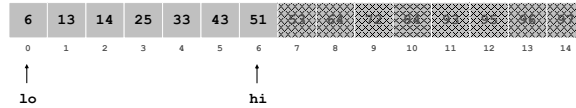
Figure 3: Step3

Ex. Binary search for 33.

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

↑ lo   ↑ mid   ↑ hi

Figure 4: Step4

Ex. Binary search for 33.

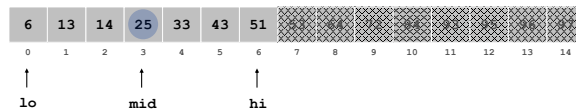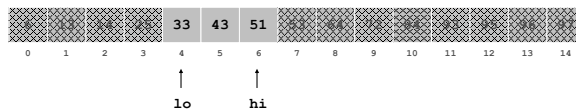| 6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

↑ lo   ↑ hi

Figure 5: Step5

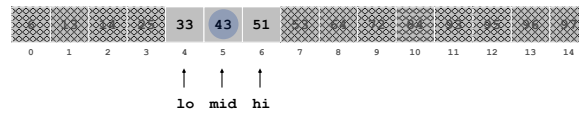Ex. Binary search for 33.

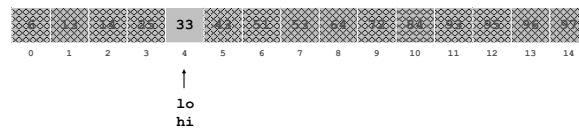Figure 6: Step6

Ex. Binary search for 33.

Figure 7: Step7

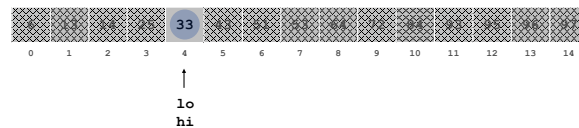Ex. Binary search for 33.

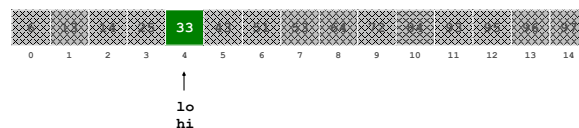Figure 8: Step8

Ex. Binary search for 33.

Figure 9: Step9

You will write a program that reads an input file X.txt which stores student names and their cities separated by a colon(:) and prints out the search value and its city or - your value is not in your file - if the search value is not found

Hint: You have to give sorted list in your binary search function. Its mean you must sort the list which you read from the input file.

You are forbidden to use built-in sort function. You must implement your own function. You are free while implementing the sort function (Insertion, Bubble, Merge, etc.)

While searching the value you have to write each step of your code into file. (snapshot) .

Problem4: Plotting on Diabetes Data Set

Plot 1

Create plot that displays the distribution of the age of instances whose Diastolic blood pressure greater than 90 (border included) and shows the followings in that plot. Save as Fig1.pdf

- The maximum rate of the number of pregnancies by age (#pregnancy/age)

- The minimum rate of the number of pregnancies by age (greater than zero) (#pregnancy/age)

- The maximum rate of the body mass index by age (Body mass index/age)

- The minimum rate of the body mass index by age (greater than zero) (Body mass index/age)
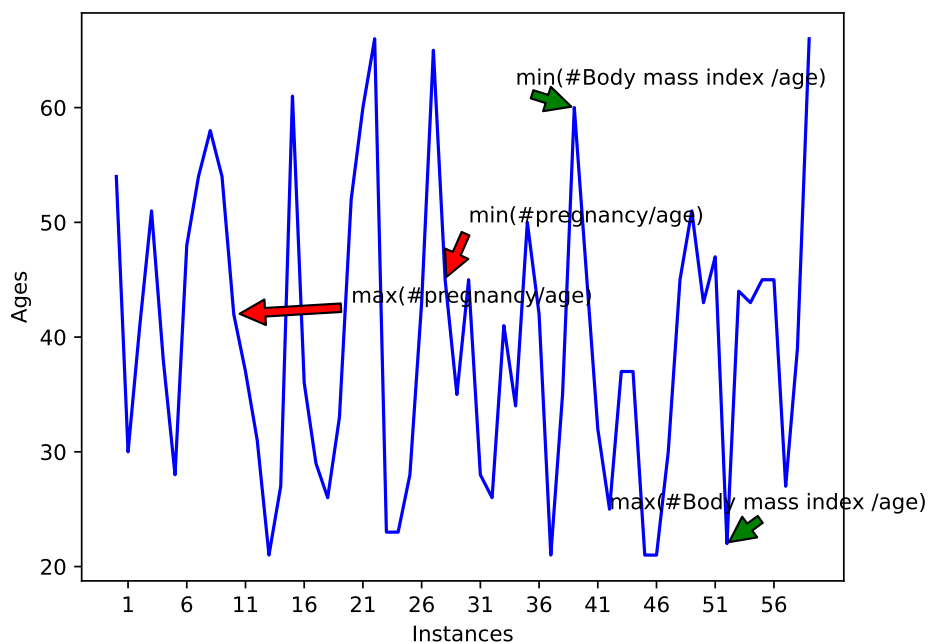

Figure 10: plot1

Plot 2 :

Create bar plot that displays the number of patient .The X-axis must be formed in year..
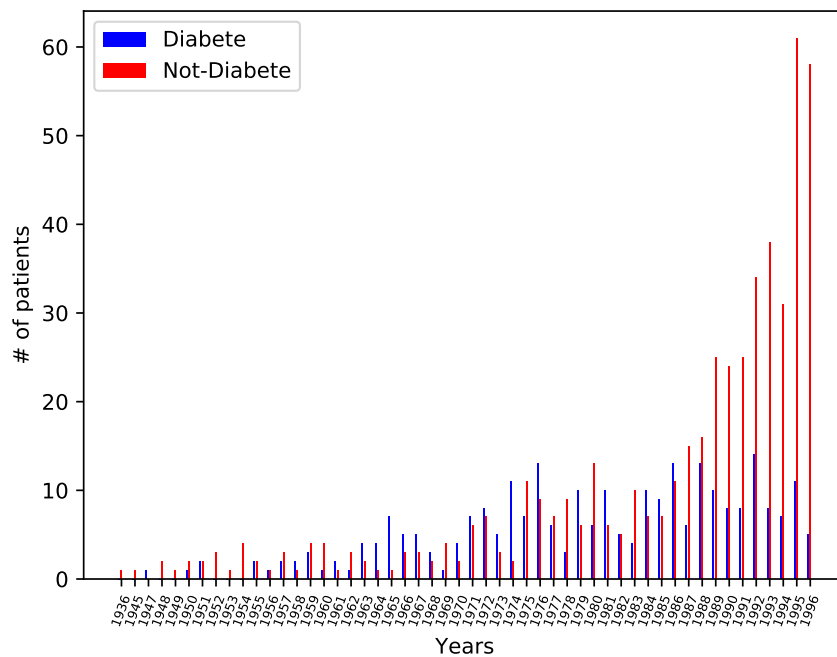Save as Fig2.pdf

Figure 11: plot2

Do not show the figure window but save.

- Do not miss the submission deadline.

- Save all your work until the assignment is graded.

- You can ask your questions via Piazza and you are supposed to be aware of everything discussed on Piazza.

- You must submit your work with the file hierarchy as stated below:

$\rightarrow$ <Assignment5-1.c >
$\rightarrow$ <Assignment5-2.py>
$\rightarrow$ <Assignment5-3.py>
$\rightarrow$ <Assignment5-4.py>