C Programming lab
Batch2 Midterm Exam

Revision Date: September 30, 2013

Preliminaries

This exam has a total of 100 points. All tasks are all or nothing. There is no partial credit given for this test. The duration of this exam is two hours. You must put all your programs in the correct directories and submit from the specified directory or you will receive zero credit.

Create a directory named *batch2* that hangs off your `/home/IMT2013XXX` directory using 'mkdir' linux command. Create five directories inside the *batch2* directory: *harmonic*, *channel*, *visa*, *rail*, and *filter* directories. You should see the following output:

```
4    ./harmonic
4    ./channel
4    ./visa
4    ./rail
4    ./filter
4    .
24   .
```

The names must be as shown, but the order and the numbers do not matter.

Task1: Harmonic mean of a set of numbers from the command line arguments (20 points)

Develop a program in *harmonic.c* in *harmonic* directory which will take n command line arguments Your program is supposed to print the harmonic mean for the set of numbers parsed. For example, if your program is run with the following set of numbers parsed

```
./harmonic 10 4 4 5.0
```

then it should should provide the following output.

```
Harmonic mean: 5.00
```

The harmonic mean calculation is shown from the below formula:

$$harmonic - mean = \frac{n}{\sum_{i=1}^{n} 1/a_i} \qquad (1)$$

Here n represents the count of numbers parsed and a_i represents the individual numbers in the set. Remeber that your set of numbers can be real numbers as well and we will consider all positive numbers greater than zero.

Task 2: Paid channels (20 points)

Write a program considering your code will be ported to embedded set-top box controller. Embedded system development represents using minimum number of variables in your code and thus minimizing the memory required to run program. Also you have to chose the variable according to the size required. If the number of variables is more than the minimum required, your solution will not be evaluated.

Write a program in a file *channel.c* in *channel* directory. Your program should take one input from the user. The TV-Channels format in the set-top box machine is shown below:

```
|Sony(2-bits)|Star(3-bits)|Zee(2-bits)|NDTV(1-bit)|AAJTAK(1-bit)|CNNIBN(1-bit)|
```

The items format suggests that within their individual width format, the user can select the following TV channels

```
Sony
0: SonyTV
1: SonyPix
2: SetMax
3: SonyMix

Star
0: StarPlus
1: StarWorld
2: StarGold
3: StarMovies
4: StarUtsav

Zee
0: ZeeTV
1: ZeeClasic
2: ZeeCafe
3: ZeeCinema

NDTV
0: NDTV24X7
1: NDTVProfit

AAJTAK
1: AAJTAK

CNNIBN
1: CNNIBN
```

The price of the tv-channels is supplied by the command line argument and is of following format:

```
|Sony(6-bits)|Star(6-bits)|Zee(5-bits)|NDTV(5-bits)|AAJTAK(7-bits)|CNNIBN(3-bits)|
```

Assuming that we have same prices for all channels within the same group. i.e Sony, SonyPix, SetMax, SonyMix will have same price in group Sony group. Also StarPlus, StarWorld, StarGold, StarMovies, and StarUtsav will have same price and so are the other items in the same group.

Based on the price command line argument and user inputs on selection of items, your program should print the tv-channels chosen, and total price of all channels chosen.

For example if the user input is (11 001 10 1 1) in binary i.e. 411 and price argument is (111110 111110 11000 11000 1111111 111) in binary i.e 4226573311

```
./channel 4226573311
```

then your program should print the following:

```
Channels selected:
SonyMix
StarWorld
ZeeCafe
NDTV24X7
AAJTAK
CNNIBN

Total price:
306
```

Task 3: Visa interview (20 points)

Write a complete program in *visa.c* in *visa* directory to determine the first and last candidate who will be interviewed from the given list. The first and last candidate will be selected based on the interview time available in the list. Your program should also state the type of visa the interview will be conducted for first and last candidate. The given list is provided to your program as file-name via command line argument. The list of interviewers is given in the following format:

```
<Name> <InterviewTime> <VisaType>
```

One such list in *test.dat* is given below:

```
JohnLusth 14.20 H1B
SusanBurkett 4.20 H1B
MichelleDubose 20.20 B1
StevenSpiegel 24 F1
```

On running the above test case as shown below,

```
./visa test.dat
```

should provide the following output:

```
First candidate: SusanBurkett H1B
Last candidate: StevenSpiegel F1
```

Assume that we will not have more than one candidate with same interview time.

Task 4: Railway stations (20 points)

Write a complete program in *rail.c* in *rail* directory to trace the stations starting from nearest railway station to nearest destination station. The list of stations where the trains pass by is given to you in the order of its reaching the station. The list is supplied via command line argument in the following format.

```
<Station-Name> <X-coordinate-location> <Y-coordinate-location>
```

One such example in stations.dat file is given below:

```
Thane 100 23
Mulund 100 35
Bhandup 100 55.5
Ghatkopar 100 75
Kurla 100 90.3
Sion 100 93
Byculla 100 120
CST 100 140
```

The starting and destination positions will be supplied as command line arguments while running the program in following format:

```
./train <file-name> <startX> <startY> <destinationX> <destinationY>
```

If your program runs as shown below:

```
./train stations.dat 90 21 103 129
```

then your program should provide the below output:

```
Thane->Mulund->Bhandup->Ghatkopar->Kurla->Sion->Byculla
Distance travelled: 116.68
```

Remeber that the distance travelled will include the distance travelled by the train as well as distance from starting positions to nearest station and distance from nearest station to destination positions. Note that your program will be tested with different test files and different starting and destination locations.

Task 5: Band pass video filter (20 points)

Develop a Band pass filter program in file filter.c in *filter* directory. Assuming that just like RRA-audio format, one of your classmate has developed an RRA based video format. One such RRA based video format (in.RRV) is given below.

```
RRVIDEO
samples: 5
createdBy: IMT2013XXX
%%
10
-20
30
-40
50
```

Assuming that the video has noise in between two cutoff values and hence given the cutoff values, your filter should be able to remove the sample values lying in between the two cutoff values (inclusive), retaining all other sample values. For example if the cutoff values parsed is 20 40, then the new filtered video file is as follows:

```
RRVIDEO
samples: 2
createdBy: IMT2013XXX
```

```
%%
10
50
```

Note that the samples tag in the header is changed to reflect the exact samples retained by this filter. Your program should be developed to accept threshold value and input VIDEO file from the command line arguments and print the filtered RRVIDEO file into the standard output console. Your program will be tested in the following format:

```
./filter <cutoff-value1> <cutoff-value2> <RRV-filename>
```

Note that cutoff-value1 and cutoff-value2 can be lower or greater. This means that if your program is tested with the following arguments as shown below:

```
./filter 20 40 in.RRV
or
./filter 40 20 in.RRV
```

your program output should be same.

Submitting your midterm exam

First, move into your `~/batch2` directory. Then run this system command:

```
    du -a
```

and check whether all your files are located in appropriate directory.

Once you are done and ready to submit, stand up and wait for your instructor to come to your place before submission. You need to submit your midterm exam only infront of the instructor. In case if you submit without instructors presence, your submission will be unaccepted. Also sign the attendance sheet with the time of submission before leaving the room. Submit your solutions while in the `~/batch2/` directory with the command:

```
submit clab mr batch2 <your-iiitb.org-email-address>
```

Write your name, email-id and roll numbers in the question paper as well. Give your exam question paper back to your instructor. Have a nice break!!