The motive of this program is to create a test to ascertain if a person, with repeated and assisted trials, is able to count a second as accurately as possible. The test will consist of the software showing the user a count down from 5 every second, to 1 (the display stops at 1) and the user should press a button when he/she anticipates that it would have become 0. The time recorded is the time from the actual zero that the button is pressed. For example, if the user presses the button say, 24 msec before it is actually zero, the time for that trial is recorded as -0.24 (in sec). Similarly if he/she presses it 17 msec after the actual zero it will be recorded as 0.17 (in sec). There will be 10 such trials and at the end, the time recorded at each trial is displayed in a graph along with other statistics which is described below.

The following are the requirements of the program.

1. The user interface contains only a button **B**, a numeric/label to display a countdown **C** andanothernumeric/label to display the trial number **T**. It should also contain a button saying “Start Test” which should disappear once the user clicks on it to begin the test.
2. Each trial will consist of counting down (in unit steps) from 5 every second while displaying it in **C**.
3. **C** is updated only till the count reaches 1.
4. If the user presses **B** say at any time **t**, then either
   1. **t** will be the time remaining (negative and in seconds, up to two decimal places) if the counthasn’t become 0
   2. or **t** will be the time elapsed after the count reached 0(positive and in seconds, up to two decimal places).
5. Once **B** is pressed, **t** is recorded and stored in memory and the next trial is started.
6. There should be 10 trials and the user should be shown the current trial number in **T**.
7. At the end of 10 trials, the user should be thanked for participating in the test.
8. Once the user acknowledges, a new panel should be opened (no other panel/popups should be visible) which should contain a graph with the number of trials along X axis and **t** for each trial along Y axis. Note that the graph should be properly labeled.
9. Below the graph, display the following
   1. Best time (either positive/negative) along with trial number.
   2. Average time (average of the absolute value)
   3. Best time ever along with the user’s name who created it and his/her trial number
   4. Best average time ever (either positive/ negative) along with the user’s name who created it
10. It goes without saying that the 9 c) and 9 d) is stored in a separate file, called “hs.dat”.
11. If the current user has created a new record i.e. 9 c) and 9 d), then congratulate the user and request for the name of the user. Accept only if the text entered is not empty, else request again.
12. Store the new record in the file and refresh the panel opened in 8) with the new record detail.
13. The user should be able to exit the program anytime (with a confirmation to quit) by clicking the upper left cross only (no separate quit button).

The above features are mandatory in the program and these will carry the essential points. However it is possible to score additional points, that could cover up for any mistakes you might make, by implementing as many as additional features as you can from this list (only). Implementing all will attract special mention and additional bonus, which is yet to be decided!

1. The countdown in 2) above should be a random number between 3-10 (instead of 5) for each trial. The user should be able to select a checkbox saying “Random count” before beginning the test and it should also disappear when the user begins the test by pressing “Start test” (refer to 1) above).
2. The user’s name entered in 11) should be accepted if and only if it contains at least 3 alphanumeric characters, no space(s) in the beginning or end and no special characters.
3. A new trial is started if the user doesn’t press **B** within 2 seconds after the countdown of that trial becomes 0. No indications is necessary.