OS Lab Test 1 **March 17, 2023** **9.15 - 10.30 AM**

Venue: CSE department Software Lab (Main building and Annex building)

- Every student will have to take the test individually.
- You are not allowed to keep any notebook, mobile phone, etc. during the test. As per IIT Kharagpur rules, mobile phones or any other gadget are not allowed during any exam, even in switched-off mode.
- Plagiarism in any form will be severely penalized. If the programs of two or more students are found similar, all of them will be awarded zero (minimum punishment) irrespective of who has copied from whom. There can be larger penalties depending on the situation.
- It is your duty to write your codes in a way so that they can be understood by those who will grade your test. Use meaningful variable names, write short comments if needed.
- There will be no Internet access during the test. Internet access will be provided for 5 minutes after 10.30 am, during which the programs need to be uploaded to Moodle.

Upload instruction:

- **Put all of your files (problem1.sh, problem2.c, problem3.c, instruction.txt) in a single zip file named <your-roll-no>.zip**
- **Upload the zip file in Moodle, there is an assignment called "lab-test-1"**

Problem 0. **[No marks without this]**

Create an instruction file named "instruction.txt" with instruction on how to compile / run your code. We will follow this file verbatim to compile and run your code while grading.

Problem 1: **[10 marks]**

Write a shell script which takes three numbers x, y and m as input. Check if all of x, y and m are positive non-zero integers, if they are not, print "invalid input" and exit. If x, y and m are positive non-zero integers then output $\left(x^{y^2 mod\ m}\right) mod\ m$ as output.

E.g., if x = 5, y = 3, m = 2 is given as input then output will be $5^{3^2 mod\ 2} mod\ 2 = 5^{9\ mod\ 2} mod\ 2 = 5^1 mod\ 2 = 5\ mod\ 2 = 1$.

Recall that "a mod b" means the remainder when a is divided by b.

IMPORTANT: The shell script should be in a single file named "*problem1.sh*"

Problem 2: **[15 marks]**

Write a c code which runs a while loop and waits for an input character. For any input it just prints the input character (and a newline) to terminal and wait for next input, except for 3 cases:

- If the input is not a letter (upper or lowercase), the code prints "Do you speak-a my language?" and waits for next input
- If **ctrl + c** is pressed the code prints "I am unstoppable!" and waits for next input
- If "x" is given as input the code exits after printing "Valar Morghulis"

Example run:

```
> ./problem2
> a
> a
> 1
> Do you speak-a my language?
> [press ctrl + c on keyboard]
```

> I am unstoppable
> x
> Valar morghulis (and exits)

IMPORTANT: The c code should be in a single file named *"problem2.c"*

Problem 3: [25 marks]

Write a c code which does the following:

- Takes input m and n, two positive non-zero integers
- Create a two-dimensional matrix of size m x n, named M
- Fill each element of the matrix M with an integer randomly chosen from 1 to 1000
- Print "Random matrix M is created"
- Create two arrays A and B, each of constant size 1000.
- Print "Shared arrays A and B are created"
- Print all the value of matrix M (in a m x n format) in the main thread.
- Spawn three "*order threads*" and one "*chaos thread*".
- Print "I am order" and "I am chaos" in order threads and chaos thread respectively.
- The order and chaos threads will do the following.
- Chaos thread
    - The *chaos thread* will choose a random cell from the matrix M and update the value with a random number from 1 to 1000.
    - It will print "Chaos: updated element at cell i x j with value k" (print appropriate values of i, j, k in the printed message)
    - Then *chaos thread* will store the changed row number i in array A and in array B mark the corresponding index with a macro *UNPROCESSED*.
    - After each change the chaos thread will sleep for two seconds.
    - After thirty such updates chaos thread will print "CHAOS ENDS" and exit.
- Each of Order threads
    - Each *order thread* will check the shared array A and B for new unprocessed entry. If there is any, then one order thread will lock the shared arrays A and B, read one changed row number, mark it with a macro PROCESSED and then unlock the arrays A and B.
    - It will print "Order: detected updated element at row i" (print appropriate value of i in the printed message)
    - Next the order thread will lock the changed row in the two-dimensional matrix M, sort the changed row in increasing order and unlock the row of M.
    - It will print "Order: row i is sorted now"
    - It will then print "older row i:" [row values before sorting, space separated]
    - It will then print "new row i:" [row values after sorting, space separated]
    - Then the order thread will go on checking the shared arrays A and B for more unprocessed element.
    - After processing thirty such updates order threads will exit.
- Finally. print all the value of matrix M (in a m x n format) in the main process and exit.

IMPORTANT: The c code should be in a single file named *"problem3.c"*

------------------ LAB TEST 1 END OF QUESTION: BEST OF LUCK ------------------