Section 16 Mid Semester Examination

PDS Lab **SET- I** **11.09.2018**

Instructions:

*Create a sub directory named as **MidSem**.*

Give the name of the programs as <p>_1.c, <p>_2.c, .. etc. for the problem 1, 2….., respectively. Here <p> implies the part number. For example, Part-A

*Store all the programs under this assignment in the directory **MidSem**.*

*Zip the entire directory **MidSem** and rename it as **<R>_ MidSem _tar.gz**. where <R> denotes your Roll No.*

You should upload your zipped file to the Moodle course web page latest by 11:55 hrs.

Part-A

1. Read the input values from the keyboard, calculate the following expression and print the final result on the display screen.

$$p = \frac{1 - \sqrt{\frac{\rho}{(R-T)^2}}}{\left(1 + \frac{\sigma}{1 + \frac{1}{\sigma . n}}\right)^2}$$

Given that $R = 8.320$

$T = 373.46$

$\rho = 1.432 \times 10^{-5}$

$n = 6.032 \times 10^{23}$

$\sigma = -5.26 \times 10^{-11}$

Test case:

Consider the above-mentioned data as the test case.

2. Read a set of numbers (including both positive and negative values) from the keyboard. Find the minimum and maximum numbers you have entered. Don’t use any array.

Test cases:

```
-3   2   5   -5   6
-9   -3   -7   -2   -1
10   8   6   2   5
55
```

3. Define an array say `y` of characters of maximum size say 20. Read a number say `n<20` from a user and then ask the user to enter characters. You should continue reading characters from user until `n` characters are entered in such a way that all characters in `y` are unique and they are in alphabetical order. You should ignore entering a character if it is

already present in the array or it does not satisfy the alphabetical ordering.

Test cases:

```
n = 4
a  e  b  i  c  o
```

Part-B

4. Read a set of numbers and store them in an array. Print the array after reading all the numbers. Arrange the numbers in the reverse order of their entries. You should not use any extra array for this. Print the whole array after reversing their positions.

   Test cases:

   ```
   10  20  30  40  50
   11  44  33  22
   ```

5. Read two sets of numbers from the user and then store them in two arrays say `x` and `y`. Find all numbers which are not common in `x` and `y`.

   Test cases:

   ```
   A: 30  5  7  9                A: 1  2  3  4
   B: 1  3  5  7  11             B: 5  6  7
   ```

Part-C

6. Define a recursive function to check whether a string is a palindrome or not. The function should return 1 if a string is palindrome else return 0. Call the function from the `main()` for n-times (the value of n should be read from the user) . In each loop, you read a string, print the string, call the recursive function and then print the string, if it is palindrome.

   Test cases:

   ```
   n = 4
   MADAM
   TOP
   MALAYALAM
   BANANA
   ```

---*---

Submission instruction

Zip your ***Midsem*** directory. Upload your zip file into the Moodle server.