tcg crest
Inventing Harmonious Future

Institute for Advancing Intelligence, TCG CREST
(TCG Centres for Research and Education in Science and Technology)

Introduction to Computer Programming and Data Structures
Assignment 05

Maximum Marks: **100** Submission Deadline: **2022-Sep-20**
Bonus: **20** – programming style and efficiency

Assignment problem # AP0501

- Problem: Given a floating point number a, print its binary representation.
- Input: a
- Output: binary representation of a

[10]

Assignment problem # AP0502

- Problem: Given an integer a, test if at least three of last five bits of a are 1.
- Input: First line: n /*an integer*/
- Output: yes/no

[10]

Assignment problem # AP0503

- Problem: Frequency analysis: Suppose you are given a dictionary of English words. Let $\mathbb{A} = \{a, b, \ldots, z\}$ be the set of lowercase letters.

  1. Compute the non-case-sensitive frequency frequencies of each letter. Let $fr_a, fr_b, \ldots, fr_z$ be such frequencies. Then, for each $a \in \mathbb{A}$, non-case-sensitive frequency fr_a counts occurrences of both a and A.
  2. Compute the non-case-sensitive frequencies g_{ab} for each pair of letters $ab \in \mathbb{A} \times \mathbb{A}$.

- Input: path of the dictionary (source file download link)

- Output: List of letter/pair of letters and its frequency, separated by a space. Example

  - $a\ fr_a$
  - $b\ fr_b$
  - $\vdots$
  - $z\ fr_z$
  - $aa\ fr_{aa}$
  - $ab\ fr_{ab}$
  - $\vdots$
  - $zz\ fr_{zz}$

[25]

Assignment problem # AP0504

- Write a program in C to print frequencies, in ascending order, of all unique elements in a given 2D array.

- Input matrix must be taken from the file titled "input_matrix.txt".

- The 2D input matrix is kept in the input file as follows
  $n\ m$
  $a_{11}\ a_{12}\ \ldots\ a_{1m}$
  $a_{21}\ a_{22}\ \ldots\ a_{1m}$
  $\vdots$
  $a_{n1}\ a_{n2}\ \ldots,\ a_{nm}$

[20]

Assignment problem # AP0505

- Problem: *Linked-list and string*: Suppose you are given a list of strings.
- Create a singly link list. Each node in the list should contain a character pointer to a word and a link to the next node.
- Take the strings from the file "list_of_strings.txt" one by one, create a node, add the string in that node, and append the node in front of the link list.
- Write a function to print the strings stored in the nodes of the linked list.
- Rearrange the nodes so that strings are sorted in ascending order in the link list. Use `strcmp`() from `string.h` to compare strings.
- Write a function in C to check whether a given string is a sub-sequence of at least one of the string from the dictionary.
- Write a function that, given a string of at least length 5, deletes all the nodes containing the given string as substring.
- Print the values in the linked list after every operation.
- Write user menu/choice accordingly.

[35]