Assignment-1

Data Structures and Algorithms
Pointers, Abstract Data Types, and Linked Lists

Due Date: 28 March, 2023

Important Notes

- For question 1 and 2, all function mentioned must be defined with **exact** function name and function parameters and return type (we would run scripts to check this, so match the case of alphabets also)

1 Pointer Warmup

Pointers is a very important concept in C programming. Let us begin with a few warmup questions.

1.1 File Structure

- `functions.h` - Which has the function prototypes.
- `functions.c` - Which implements the functions defined in 1.2.
- `main.c` - Which must use the functions and perform various operations on input and produce output as specified later.

1.2 Functions

- `void reverseString(char* str, int length)`: takes in a pointer to a string and its length, and reverses the string in-place.
- `int* uniqueElements(int* arr , int length)`: takes in a pointer to an array of integers and its length, and returns a pointer to an array of integers that contains only the unique elements of the original array, without using any library functions except `malloc`.
- `char* compressString(char* str, int length)`: takes in a pointer to a string and its length, and returns a pointer to a string that is a compressed version of the original, where each character is followed by the number of times it appears consecutively in the original string. For example, "**aaabbc**" would be compressed to "**a3b2c1**".
- `int** transpose(int** matrix , int NumRow , int NumCol)`: takes in a pointer to a matrix of integers and its dimensions, and returns a pointer to a matrix that represents the transpose of the original matrix.

1.3 Input and Output

The first line contains T, the number of OPERATION's that need to be done. $1 \leq T \leq 100$

The first line of an OPERATION consists of a string indicating the OPERATION name.

The next few lines contain the required input data according to the table i.e.

Operation name	**Corresponding function**
OPER1	*reverseString*
OPER2	*compressString*
OPER3	*uniqueElements*
OPER4	*transpose*

OPERATION	INPUT FORMAT	CONSTRAINTS	OUTPUT FORMAT
OPER1	First line contains an integer n indicating the length of the string. Second line contains the string	$1 \leq n \leq 10^4$	String (Reversed)
OPER2	First line contains one integer n indicating the length of the string. Second line contains the string	$1 \leq n \leq 10^4$	String (Compressed)
OPER3	First line contains an integer n indicating the length of the array. Second line contains n space separated integers.	$1 \leq n \leq 10^4$ $1 \leq Arr[i] \leq 10^6$	All unique elements in a line, space separated
OPER4	First line contains a 2 integers R and C indicating the number of rows and number of columns respectively. Following R lines contains C space separated elements	$1 \leq R, C \leq 10^2$ $1 \leq Mat[i][j] \leq 10^6$	The transposed matrix. Each row should be printed on a new line

1.4 Example Test Cases

Input	Output
4	
OPER1 10 abcdefghij	jihgfedcba
OPER2 12 aacceeggiiaa	a2c2e2g2i2a2
OPER3 10 1 1 2 4 5 9 9 1 6 8	1 2 4 5 9 6 8
OPER4 3 4 1 2 3 4 2 3 4 5 3 4 5 6	1 2 3 2 3 4 3 4 5 4 5 6

2 Array Warmup

Arrays are a very important concept in programming as they let us efficiently store data in the linear format. Let us begin with a few warmup questions.

2.1 File Structure

- `functions.h` - Which has the function prototypes.
- `functions.c` - Which implements the functions defined in 2.2.
- `main.c` - Which must use the functions and perform various operations on input and produce output as specified later.

2.2 Functions

- `int* IntersectionArray(int* Arr1 , int* Arr2, int lenArr1, int lenArr2)`: takes in two pointers to arrays of integers and their lengths, and returns a pointer to an array of integers that contains the intersection of the two arrays i.e. common elements in both arrays. Do not use any library function other than `malloc`.
- `int countCharOccurrences(const char* str, int length, char ch)`: takes in a pointer to a string, its length and a character and returns the count of the number of occurrences in the string.
- `char findFirstNonRepeatingChar(const char* str, int length)`: takes in a pointer to a string and its length, and return the first non-repeating character in a given string and return the character.
- `char* findLongestCommonPrefix(char** strs, int numStr, int maxLen)`: takes in a array of strings, number of strings and the maximum possible length. Returns the pointer to the longest common prefix among all the strings in the array.
- `int* MaxMin(int* Arr, int lenArr)`: takes in a pointer to an array of integers and its length, and returns a pointer to an array of integers that contains the indices of the elements that are greater than all the elements to their left and smaller than all the elements to their right.

2.3 Input and Output

The first line contains T, the number of operations that need to be performed. $1 \leq T \leq 100$.

The next line consists of a string indicating the OPERATION to be executed.

The next few lines give the data needed for the operation i.e.

Operation name	**Corresponding function**
OPER1	*IntersectionArray*
OPER2	*countCharOccurences*
OPER3	*findFirstNonRepeatingChar*
OPER4	*findLongestCommonPrefix*
OPER5	*MaxMin*

OPERATION	INPUT FORMAT	CONSTRAINTS	OUTPUT FORMAT
OPER1	First line contains two integers n_1 and n_2 indicating lengths of first and second array. The following two lines contains $n1$ and $n2$ integers respectively.	$1 \leq n_1, n_2 \leq 10^4$ $1 \leq Arr1[i] \leq 10^6$ $1 \leq Arr2[i] \leq 10^6$	The common elements in a single line, space-separated
OPER2	First line contains a string. The second line contains a character	$1 \leq n \leq 10^4$ $Arr[i]$ can be a digit or lower or uppercase Alphabet	The count of the number of occurrences of the character in the string
OPER3	One line containing a string.	String length $\leq 10^4$	The first non-repeating character in the string. If there are none print a empty line
OPER4	First line contains one integer n indicating the number of strings. The next n lines contain a integer n_i followed by a string each.	$1 \leq n \leq 10^2$ length of each string $\leq 10^2$	The longest common prefix of all the strings
OPER5	First line contains one integer n indicating the length of the array. The second line contains n space separated integers.	$1 \leq n \leq 10^2$ $1 \leq Arr[i] \leq 10^6$	Indices of elements that are greater than all elements to their left and smaller than all elements to their right in a single line, space-separated

2.4 Example Test Cases

Input	Output
5	
OPER5 10 1 4 8 7 5 9 17 11 40 10	1 5
OPER1 6 7 1 2 3 4 5 6 7 9 1 4 8 2 3	1 2 3 4
OPER2 helloworld l	3
OPER3 lleetcode	t
OPER4 3 flower flow flight	fl

2.5 Notes

- In Max-min, for first element just consider whether element after it is greater and for the last element just consider whether elements before it are lower.
- In Intersection-Array, it is not necessary that the output array is sorted. Any order is accepted.