Programming and Data Structures Laboratory | 2021-22 Autumn semester, Section 20
Assignment 8 | February 22, 2022

Submission instructions

* Submit one .c file for each part of the assignment. Each of your .c files should be named as:

<your roll number>_A<assignment number>_<part number>.c

For instance, if your roll number is 21CS10023, and if you are presently doing Assignment 8 which has 3 parts, then you should submit 3 separate .c files named as:
21CS10023_A8_1.c 21CS10023_A8_2.c 21CS10023_A8_3.c

***** Submissions must be through the course Moodle, before the end of Lab session (11:55 AM). Late submissions will be penalized / not accepted.
***** Try to think how you can solve a problem efficiently. A program that solves the problem efficiently will get more credit.

---

1. [10 Marks] Declare an integer variable `n`. Read its value from the keyboard and dynamically allocate an integer array `a[]` of length `n`. Fill up the array by `n` integers value from the keyboard. Print the elements of `a[]` in the terminal.
Write a function that takes the address and the length of an array as arguments, and shifts its content to the right by one place with wrap around, as shown in the example. Call this function from the `main()` with `a[]` and the length of the array as input. Print the array `a[]` in the terminal after the function call. Do this one more time.
Example output:

```
Enter number of elements: 5
Enter the elements: 7 3 6 1 4
a[] = 7 3 6 1 4
After function call:
a[] = 4 7 3 6 1
After function call:
a[] = 1 4 7 3 6
```

2. [20 Marks] Declare a 2D integer array a[][] and define its size (#rows and #columns) taking user input. Write a function to dynamically allocate the space for a[][]. Fill it up by integer values from the user. Write another function to find its saddle point(s) and print on the terminal. If there is none, then print that message.
An element is a saddle point if it is the minimum in its row as well as the maximum in its column. Similarly, if an element is the maximum in its row as well as the minimum in its column, then also it is a saddle point.
Assume that the user enters the matrix such that the minimum or maximum values in any row or column is **unique**. That is – the minimum or maximum value does not repeat inside the same row or column.
Example output:

```
Enter number rows and columns: 3 3
Enter the array (one row at a time):
3 1 7
4 3 8
1 2 5
Saddle point is a[1][1]=3
Saddle point is a[2][2]=5

Enter number rows and columns: 5 5
Enter the array (one row at a time):
```

```
23 25 19 55 11
80 50 69 55 70
11 15 45 95 19
24 25 33 50 32
62 35 22 78 35
Saddle point is a[1][1]=50
Saddle point is a[3][3]=50

Enter number rows and columns: 3 3
Enter the array (one row at a time):
2 5 7
1 2 5
7 3 4
No saddle point exists for the given input matrix.
```

3. [20 Marks] "Favorite books" in a library are those which have been read by all those members who have read at least 50% books. For this, consider a library with m members and b books. The values of m and b are given as input. Moreover, the library also maintains an integer matrix S where S[i][j] = 1 if the i-th member has read the j-th book, and S[i][j] = 0 otherwise. Dynamically allocate the m-by-b 2D array S and fill it up using data from the user.
(a) Find and print the members who have read at least 50% books.
(b) Find and print the favourite books.
Write appropriate functions to solve the problem.
Hint: Try to resuse some part of the programs you wrote in the previous problem.

Example:

```
#members = 3. #books = 5.
S matrix:
1 1 1 0 0
0 0 1 1 0
0 0 0 0 1
Members reading at least 3 books: 1.
And the favorite books are: 1, 2, 3.

#members = 7. #books = 12.
S matrix:
1 1 0 0 0 1 0 0 0 0 0 1
0 0 1 1 1 0 1 1 0 0 0 0
1 1 1 1 0 1 0 1 1 1 0 1
0 0 0 1 1 1 0 0 0 0 0 0
1 1 1 1 0 0 0 1 0 0 1 0
0 1 1 1 0 1 0 0 0 0 0 1
0 1 1 0 1 0 1 1 0 0 1 1
Members reading at least 6 books: 3, 5, 7.
And the favorite books are: 2, 3, 8.
```