

**Indian Institute of Technology Bombay, Mumbai**  
**Department of CSE, Kanwal Rekhi Building**  
**CS101 – Computer Programming**  
**Autumn Semester 2014-2015**

**Special Makeup Lab 4 – Handout**  
**Tuesday, 28 October 2014**

**General Instructions:** *Please read the instructions carefully before proceeding further.*

Read the entire lab handout carefully and understand it.

This lab requires you to **download, execute, and submit** one program today before you leave the lab.

The code snippet (splab4.cpp) is uploaded under '**Special Lab**' on course page on <http://www.cse.iitb.ac.in/~cs101/labs.html>

- I. **Download this splab4.cpp** file: Right click on **splab4.cpp** – > click '**Save link as**' – > Go to the desired directory and click '**Save**'.
- II. Load this .cpp in Code::Blocks
- III. Fill up the necessary blocks of the code
- IV. Compile and execute the splab4.cpp file
- V. Submit **only a single file** (splab4.cpp). The instructions for submitting/uploading a file are given below:
  - Goto <http://www.cse.iitb.ac.in/~cs101>
  1. Click on '**Lab Assignment Submission**' link.
  2. Write your roll number in the text box.
  3. Enter your CC (Computer Centre) LDAP password for authentication purposes
  4. Click the '**Choose File**' button.
  5. Browse through your directory by navigating to the folder in which you have created the project. Select the program i.e. '.cpp' file, from your project directory.
  6. Click the '**Submit**' button.
  7. A new page will open with the message '**Upload Successful. Click here to go back**'
  8. Perform these steps (1 to 7) for all the programs that you have written.
- VI. Please note that your IP address is being logged. So, only the lab assignment submissions made from the lab will be evaluated. **Submissions made from the hostel or outside the lab will NOT be considered.**
- VII. Since the .cpp file that you submit will be evaluated using an auto-grader, **DO NOT** write any cout statements, else your code **may not pass the auto-grader checks**

**Note:** Copying code from others amounts to violation of the honor code, and if detected, will lead to severe penalties, which could include award of the **FR** grade.

**Objective:** In this lab, you are required to solve practice problems based on arrays

## Programs to be Submitted Online

The program splab4.cpp has 3 blocks of code that needs to be filled up. The first is Q1, second is Q2, and third is Q3 (Bonus Question and optional).

For Q1, you need to fill in the code within the BEGIN and END statement of the first empty block, which is 'DONT\_ERASE\_02\_01'

For Q2, you need to fill in the code within the BEGIN and END statement of the second empty block, which is 'DONT\_ERASE\_03\_01'

For Q3, you need to fill in the code within the BEGIN and END statement with a 'BONUS' tag of the third block, which is 'DONT\_ERASE\_04\_01'

Take help of your TA if you have difficulty in indentifying the blocks of code.

### Question 1

An array '**marks**' contains the marks of **20 CS101 students**. A variable '**searchValue**' contains an integer number. Both, the array '**marks**' and variable '**searchValue**' are accepted from the user in the main function. These are then passed to a function '**search**'. The function '**search**' should do the following:

- 1) Find whether the number '**searchValue**' passed to this function is present in the array '**marks**'
  - 2) If yes, it should find out "how many marks/numbers present in the array '**marks**' are greater than the number '**searchValue**'. This number should be stored in a variable '**result**'.
  - 3) If the number '**searchValue**' is not found in the array '**marks**', then it should store the number '-22' in the variable '**result**'.
  - 4) The final value of variable '**result**' should be returned to the main function
- 

### Question 2

An array '**finalMarks**' contains the marks of Final exam of **20 CS101 students**.

E.g. **95, 96, 45, 67, 95, 96, 67, 43, 44, 97, 95, 67, 67, 43**

**The function 'compute' has the following parameters**

- a) **finalMarks**: Array containing final marks of CS101 students
- b) **count**: Array containing the count. This is explained later.

**The function compute should find out:-**

**a) how many students scored unique marks and store in variable 'result'.**

The value of result should be returned to the main function

E.g. If we consider the array given above, there are 3 students who scored unique marks. i.e. the number (45, 44, and 97) appears only once in the array. Store this number 3 in the variable 'result' and return this to the main function

**b) how many students scored a particular marks**

E.g. if we consider the array given above, following should be the result

**3 students scored 95**

**2 students scored 96**

**4 students scored 67**

**2 students scored 43**

**We need to achieve this in the following manner**

- i) count the number of times a particular number appears in the array.
- ii) Store the number and its count in another two dimensional array '**count**' which will be printed in the main function.
- iii) Populate the rest of the count array with the number -22

E.g.

Array 'number' = 95, 96, 45, 67, 95, 96, 67, 43, 44, 97, 95, 67, 67, 43

**count[0][0] = 95, count[0][1] = 3, Note: 95 appears three times**

**count[1][0] = 96, count[1][1] = 2, Note: 96 appears two times**

**count[2][0] = 67, count[2][1] = 4, Note: 67 appears 4 times**

**count[3][0] = 43, count[3][1] = 2, Note: 43 appears 2 times**

**count[4][0] = -22, count[4][1] = -22**

.....

**count[19][0] = -22, count[19][1] = -22**

### Question 3 – Bonus Question (Optional)

CS101 students are divided into two batches for a quiz that needs to be conducted. 250 students will appear for the quiz on Monday and another set of 250 students will appear for the quiz on Tuesday. On the day of exam, their attendance is captured. The roll numbers of the students who attempted the quiz on Monday will be stored in array '**Monday**' and those who appeared on Tuesday, their roll numbers will be stored in the array '**Tuesday**'. There is a possibility that students appeared for the quiz on both the days. In this case, their roll numbers will be present in both the arrays.

**The function 'extract' has the following parameters**

- a) **Monday**: An ineger array containing roll numbers of students who took the quiz on Monday
- b) **Tuesday**: An ineger array containing roll numbers of students who took the quiz on Tuesday
- c) **MONDAYnTUESDAY**: An integer array **initialized** to -1, to be computed in the function as explained below
- d) **MONDAYminusTUESDAY**: An integer array **initialized** to -1, to be computed in the function
- e) **TUESDAYminusMONDAY**: An integer array **initialized** to -1, to be computed in the function

**The function 'extract' should do the following**

- a) Find those students i.e. roll numbers which are present in both the arrays '**Monday**' and '**Tuesday**' and store them in the array '**MONDAYnTUESDAY**'.
- b) Find those students i.e. roll numbers which took the quiz only on Monday and not on Tuesday. i.e. find those roll numbers which are present in array '**Monday**' but not present in array '**Tuesday**'. Store such roll numbers in the array '**MONDAYminusTUESDAY**'
- c) Find those students i.e. roll numbers which took the quiz only on Tuesday and not on Monday. i.e. find those roll numbers which are present in array '**Tuesday**' but not present in array '**Monday**'. Store such roll numbers in the array '**TUESDAYminusMONDAY**'

**Note: All arrays are passed using call by reference and the function should not return any value.**

### Practice Programs (Not to be Submitted)

- 1) Write a program that accepts a 1D array 'A' of size 10 containing marks of students. Find the average of the marks stored in array 'A'.
- 2) Write a program to accept two matrices 'A' and 'B' of 3x3 size each. Declare three other matrices 'Add', 'Subtract', 'Multiply' of 3x3 size. Perform addition, subtraction, and multiplication of the two matrices 'A' and 'B' and store the result in the equivalent arrays.
- 3) Write a program to accept a matrix 'A' of 3x3 size from the user and perform transpose of the matrix 'A' and store the result in matrix 'Transpose'.
- 4) Accept two matrices 'A' and 'B' of size  $n \times n$  from the user. The size 'n' should also be accepted from the user. Multiply all the numbers of the diagonal 'A' and store in 'result1'. Multiply all the numbers of the diagonal 'B' and store in 'result2'. If  $\text{result1} < \text{result2}$ , print "Matrix A is less than Matrix B", else print "Matrix A is not valid".
- 5) Consider a 1D array of size 10. Divide the array into three equal parts. (a) left part, (b) middle part, (c) right part. Perform left shift on the left part of the array, right shift on the right part of the array and multiply the middle part with a constant value '3'. Print the modified array. If the array cannot be divided into 3 equal parts, adjust this in the middle part. i.e. the middle part can contain more elements, but the left and right part should be of equal size.

E.g. Original Array

1	2	3	4	5	6	7	8	9	10	11	12	13
---	---	---	---	---	---	---	---	---	----	----	----	----

**Modified Array**

2	3	4	1	15	18	21	24	27	13	10	11	12
---	---	---	---	----	----	----	----	----	----	----	----	----

- 6) Write a program to accept a matrix 'A' of size  $n \times n$  from the user and determine whether the matrix entered is in the upper triangular form or not.  
Refer the article for more information: [http://en.wikipedia.org/wiki/Triangular\\_matrix](http://en.wikipedia.org/wiki/Triangular_matrix)