

Section 16

PDS Lab

Lab-5

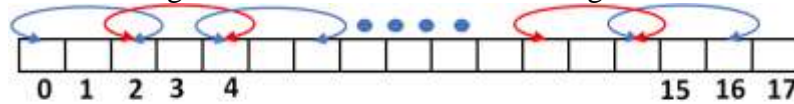
15.09.2023

Instructions:

- This lab is based on the topics: Branching and Conditionals.
- You should save each program with the file name as specified against each problem as <Lab#>_<Assignment#>_<Roll#>.c. For example, **05_01_23CS10006.c** to save Program to 1st assignment in Lab 5 with Roll Number 23CS10006
- You should upload each program to the Moodle system. Also, copy+paste your programs to the text window on the test page.
- There will be no evaluation and hence grade, if you don't submit your .c files to the Moodle server. Use **emacs** editor and **gcc command** in terminal to run the following programs.
- Document your programs meaningfully using appropriately named variable and sufficient amount of comments. Documentation and proper code indentation carry marks.
- Do not use advanced concepts like functions or structures anywhere in your code.
- **The top two lines of your programs must contain the following information:**
 - //Roll No.: <Type in your roll no.>
 - //Name: <Type in your name>

1. Write a C program to fill a single dimensional integer array of maximum size 50 with random integral numbers in the range [10, 100] by appropriately calling the rand() library function. User will first enter the number of such random numbers to be generated (maximum 50).

Display the array contents. Then, starting with the first (i.e., 0th element in the array), interchange pairs of numbers present at consecutive even positions, only if the first is larger than the second. Count the number of interchanges made. Display the contents of the array after all the interchanges and the number of interchanges.



For example, if the original array is [10,9,23,46,34,67,21], then the array after the interchanges would be [10,9,23,46,21,67,34], number of interchanges =1.

2. Write a C program to fill a single dimensional integer array of size 50 with random integral numbers in the range [10, 50] by appropriately calling the rand() library function. User will first enter the number of such numbers to be generated (maximum 50).

Display the array contents. Then, determine all the triplets that add up to 60 and display the corresponding array indices and the values stored in those locations.

3. Write a C program to read from keyboard the roll number (unsigned integer) and CGPA (float) of up to 10 students. User will first enter the number of such students for which inputs will have to be taken (maximum 10). Use two arrays of maximum size of 10 each.
 - a. Display the details read, nicely formatted
 - b. Display the average CGPA of the students.
 - c. Display the roll numbers and CGPA of all students having identical CGPA (for checking equality of CGPA, consider two digit accuracy).

4. Write a C program to read the roll number (integer), age (integer) and marks of 20 students admitted to a department. Generate random roll numbers in the range [1000,2000], age in the range [15 to 25] and marks in the range [0,100] and populate the respective arrays.. Please use three one dimensional arrays for storing roll number, age, and mark.

- a. Display the roll number, ages, and marks of all students having the same age.

Example display:

```
Roll: 1025 Age: 20 Mark:83
Roll: 3021 Age:20 Mark:45
.....
Roll 2450 Age:21 Mark:47
Roll 1975 Age 21: Mark:59
```

- b. Display the roll number, ages, and marks of all students having identical marks.

Example display:

```
Roll: 1027 Age: 23 Mark:85
Roll: 3025 Age:25 Mark:85
.....
Roll 2459 Age:21 Mark:77
Roll 1990 Age:23 Mark:77
```

- c. Sort the students according to their roll numbers and display the details nicely formatted.

Example display:

```
Roll: 1027 Age: 23 Mark:85
Roll 1990 Age:23 Mark:77
.....
Roll 2459 Age:21 Mark:77
Roll: 3025 Age:25 Mark:85
```