

Birla Institute of Technology and Science, Pilani, Rajasthan, India
Second Semester 2024-25
CS F111 – Computer Programming
Online Programming Test
SET - YELLOW

13/04/2025

Max. Marks: 90 M

Duration: 180 mins

General Instructions

- This question paper comprises three problems, whose details are described in the problem statements given on the next page. Read all the instructions and the problem statements carefully before attempting the test.
- Carefully follow the submission instructions **mentioned at the end of this page** before uploading your solutions on the **Dom Judge** portal.
- For a given problem, if you submit multiple submissions, only the latest one will be considered for evaluation. Whatever you submit on the Dom Judge portal will be considered final. **It is your responsibility to make sure that you are submitting the correct file against each problem. Also, be sure to save your files before you submit them.**
- **You are responsible for ensuring your solution is correctly submitted to the Dom Judge portal.** Later, if some student claims that he/she has submitted the solution and it doesn't appear on the Dom Judge portal, we won't entertain or listen to such issues.
- Programs that only have the input/output functions (e.g., `printf()`/`scanf()`) without the key programming logic, would **NOT** be considered for evaluation.

Instructions to attempt the test

- Create a directory in your home directory with the name **CPlabtest**.
- Download the files: "**Question Paper.pdf**" (PDF which you are reading currently), "**Q1.c**", "**Q2.c**" and "**Q3.c**" from the domjudge portal. Copy the three ".c" files into the directory **CPlabtest**. Rename your files to "**Q1_<Your_13_digit_ID>.c**", "**Q2_<Your_13_digit_ID>.c**", and "**Q3_<Your_13_digit_ID>.c**". For example, if your ID is **2024A1PS1234P**, rename Q1.c to "**Q1_2024A1PS1234P.c**".
- The above ".c" files are the files you will be working on. You will have to complete the implementation of the incomplete functions in these files according to the problem statements.
- You should **NOT** do the following in the ".c" files:
 - **change anything in the main() function**
 - **change the function parameters or the return types of the functions**
 - **change the structure definitions**
 - **create new global variables**
 - **create new functions**
- Carefully observe how each function (that you will implement) is called in the **main()** function. Also, observe the sample execution shown at the end of each question.
- Follow the submission instructions properly while submitting your solutions on the portal.
- Evaluation of your programs will be based on the following factors: **Correct Execution**, **Correctness of Logic**, and **Presentability of the Code**. Presentability includes usage of proper indentation, comments, etc.

Submission Instructions

The CPlabtest directory contains the following files: "**Q1_<Your_13_digit_ID>.c**", "**Q2_<Your_13_digit_ID>.c**", and "**Q3_<Your_13_digit_ID>.c**". Please upload these files separately to their respective submission links on the portal. Make sure to save your files before uploading. You **don't** need to convert them into zip files.

After submitting, you can download each file to verify whether they were uploaded correctly. After verifying, put an additional signature on the attendance sheet before leaving the room. Any submission without this additional signature will NOT be evaluated.

Q1. [16M] [Expected time: 30 mins]

Given an array, **arr**, of size of 50 that stores unique integer values. **arr** presently contains 40 elements in its first 40 locations, and the remaining 10 locations don't store anything meaningful. The variable **count** stores the number of elements presently stored in **arr**, which is 40. Given an incomplete function, **checkPalindrome()** that checks if the meaningful elements stored in **arr** possess a palindrome's property or not. If so, it returns 1; otherwise, it returns 0. An example of a palindrome is the word "Malayalam", which gives the same word when read in reverse. Complete the implementation of this function.

Sample Execution:

```
jagat@Jagats-MacBook-Pro-2 Set YELLOW % gcc q1.c
jagat@Jagats-MacBook-Pro-2 Set YELLOW % ./a.out
arr1 is a palindrome.
arr2 is not a palindrome.
```

PTO...

Q2. [16+16=32M] [Expected time: 1 hour]

Given an array **arr** of 20 books records, where each record consists of a **title** (string), **bookID** (integer), **publicationYear** (integer) and **edition** (int). There can be multiple books that are published in the same year in **arr**. (a) Write a function **sortRecordsByYear()** that takes the above array and sorts its elements in increasing order of the **publicationYear**, using **Selection Sort**. This function also prints the sorted array. (b) Write a function **printUniqueYears()** that prints the unique publication Years without repeating them. [Hint: Use an additional array of integers to store unique publication years.] Do NOT use functions from <string.h> library. Complete the implementation of both the above functions.

Sample Execution:

```
jagat@Jagats-MacBook-Pro-2 Set YELLOW % gcc q2.c
jagat@Jagats-MacBook-Pro-2 Set YELLOW % ./a.out
```

Sorted Book Records by Publication Year:

| | | | | | | |
|---------------|--|-------------|--|------------|--|------------|
| Title: Book I | | BookID: 109 | | Year: 2001 | | Edition: 1 |
| Title: Book R | | BookID: 118 | | Year: 2001 | | Edition: 2 |
| Title: Book J | | BookID: 110 | | Year: 2003 | | Edition: 1 |
| Title: Book S | | BookID: 119 | | Year: 2003 | | Edition: 2 |
| Title: Book A | | BookID: 101 | | Year: 2005 | | Edition: 1 |
| Title: Book C | | BookID: 103 | | Year: 2005 | | Edition: 3 |
| Title: Book E | | BookID: 105 | | Year: 2010 | | Edition: 2 |
| Title: Book B | | BookID: 102 | | Year: 2010 | | Edition: 2 |
| Title: Book F | | BookID: 106 | | Year: 2012 | | Edition: 1 |
| Title: Book K | | BookID: 111 | | Year: 2012 | | Edition: 2 |
| Title: Book L | | BookID: 112 | | Year: 2014 | | Edition: 1 |
| Title: Book M | | BookID: 113 | | Year: 2015 | | Edition: 2 |
| Title: Book D | | BookID: 104 | | Year: 2015 | | Edition: 1 |
| Title: Book N | | BookID: 114 | | Year: 2018 | | Edition: 1 |
| Title: Book G | | BookID: 107 | | Year: 2018 | | Edition: 2 |
| Title: Book Q | | BookID: 117 | | Year: 2020 | | Edition: 2 |
| Title: Book H | | BookID: 108 | | Year: 2020 | | Edition: 1 |
| Title: Book O | | BookID: 115 | | Year: 2021 | | Edition: 1 |
| Title: Book T | | BookID: 120 | | Year: 2022 | | Edition: 1 |
| Title: Book P | | BookID: 116 | | Year: 2023 | | Edition: 1 |

Unique Publication Years:

2001
2003
2005
2010
2012
2014
2015
2018
2020
2021
2022
2023

Q3. [10+10+22=42M] [Expected time: 1 hour 20 mins]

Given 10 arrays of book records (**b1** to **b10**), each of **size 4**. Each array stores four records of books found on their respective bookshelves (Shelf 1 to 10). Each book record consists of a **title** (string), **bookID** (integer), **publicationYear** (integer) and **price** (int). (a) Write a function **createPointerArray()** that takes all these ten book record arrays as parameters and indexes them in an array of pointers **shelfArr**. All the arrays should be statically allocated. (b) Write another function, **findMinPriceBook()**, that takes a book records array and finds the minimum price of all the books on that shelf. (c) Using the above function, write a function **sortPointerArray()** that sorts **shelfArr** in the increasing order of the *minimum price* of each book array stored in it, using Selection Sort.

Sample Execution:

```
jagat@Jagats-MacBook-Pro-2 Set YELLOW % gcc q3.c
```

```
jagat@Jagats-MacBook-Pro-2 Set YELLOW % ./a.out
```

Minimum prices before sorting:

Shelf 1 min price: 150

Shelf 2 min price: 300

Shelf 3 min price: 460

Shelf 4 min price: 180

Shelf 5 min price: 130

Shelf 6 min price: 720

Shelf 7 min price: 90

Shelf 8 min price: 310

Shelf 9 min price: 260

Shelf 10 min price: 550

Minimum prices after sorting:

Shelf 1 min price: 90

Shelf 2 min price: 130

Shelf 3 min price: 150

Shelf 4 min price: 180

Shelf 5 min price: 260

Shelf 6 min price: 300

Shelf 7 min price: 310

Shelf 8 min price: 460

Shelf 9 min price: 550

Shelf 10 min price: 720

==== End of document ====