ಭಾರತೀಯ ಮಾಹಿತಿ ತಂತ್ರಜ್ಞಾನ ಸಂಸ್ಥೆ ರಾಯಚೂರು
भारतीय सूचना प्रौद्योगिकी संस्थान रायचूर
Indian Institute of Information Technology Raichur

Lab - 4

Introduction to Programming (ID110)
Date: November 13, 2024
Topics: Pointers

Time: 1.5 Hr CSE'24, Semester - I Max marks: 10

Instructions:

- The lab session consists of **two programming questions**, and both of them are **mandatory**.
- External materials (e.g., notes, books) and electronic devices (e.g., mobile phones, smart watch, bluetooth) are **strictly prohibited**. Only a **blank sheet of paper** and a **pen** may be used for rough work.
- Internet usage is **not allowed** under any circumstances. Any violations will lead to **serious academic consequences**, including potential disqualification from the lab.
- Any form of **plagiarism or academic dishonesty** will be treated with the utmost seriousness and may result in severe penalties, including a zero for the lab or further disciplinary actions.
- Code must be written from scratch during the session. Pre-written code snippets or solutions will not be accepted. Use meaningful variable names and add appropriate comments where necessary.
- Upon completion, **two code files** named after **roll no.** (e.g., `"CS24B1001-Lab4-p1.c"` and `"CS24B1001-Lab4-p2.c"`) must be submitted on **Google Classroom**. Not following the naming convention will lead to **minus marking**. The submission will only be accepted if done in the presence of TA.

---

1. Write a function *rotateRight(int* arr, int size, int k)* that rotates the elements of an integer array *arr* to the right by k positions. Use only pointers to access and manipulate the array elements. Rotation should be done in place, without using additional arrays. Handle cases where k is greater than the size of the array.

   Input Format:

   - A whole number n representing the size of the array.
   - n integers separated by space representing the elements of the array. (Taking each element as input in a new line will result in negative marking.)
   - A whole number k representing the rotation factor.

   Output Format:

   - n integers separated by space representing the elements of the rotated array. (Printing each element in a new line will result in negative marking.)

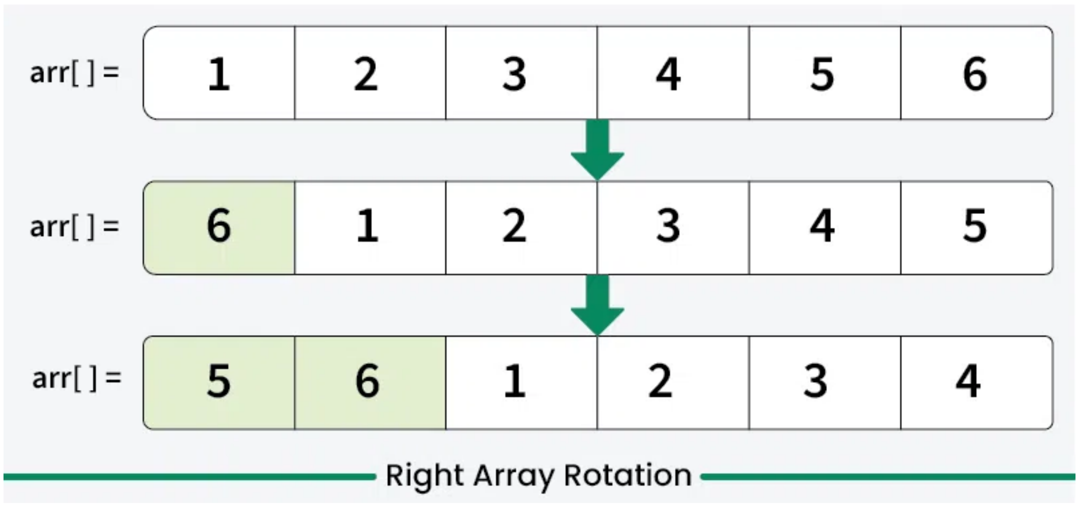

Figure 1: Right rotation of array arr[] by the rotation factor k=2

Testcases:

- **Input:**

```
6
1 2 3 4 5 6
2
```

  Output:

```
5 6 1 2 3 4
```

- **Input:**

```
9
10 20 30 40 50 60 70 80 90
4
```

  Output:

```
60 70 80 90 10 20 30 40 50
```

- **Input:**

```
4
5 6 7 8
8
```

  Output:

```
5 6 7 8
```

(4 marks)

2. Suppose you've been hired as a data analysis intern at Blinkit. You've been asked to analyze the data from yesterday's orders to find out which destination received the most parcels and some additional details. Your manager believes that this will help in optimizing delivery routes, as certain areas may consistently receive higher volumes of packages. Being an intern, you are too lazy to check the data manually so you decide to write a program that prints all the statistics about the data your manager requires.

   The company recorded all orders as numbers representing specific neighborhoods (e.g., neighborhood IDs). All the neighborhood IDs are stored in an array, *nbh*, where each element represents a delivery destination. For example, if

   ```
   nbh[7] = {101, 202, 202, 101, 101, 303, 303};
   ```

   then neighborhood 101 received three parcels, neighborhood 202 received two, and 303 received two.

   Your task is to write a function called *findMostFrequent(int* nbh, int size)* that will take in the array of neighborhood IDs and return the neighborhood ID that received the most parcels. Consider ther are no multiple neighborhoods with the same highest count i.e., only one most frequent neighbourhood exists.

   Design another function called *getOTP(int* nbh, int size)* which will simply print all the OTPs which were generated during delivery of order along with order number. The OTP for each element is it's memory address and the order number is the index of the element. You do not need to assign any OTP or order number exclusively.

   Input Format:

   - A natural number n representing the total deliveries.
   - n integers separated by space representing the elements of the array (neighborhood IDs). (Taking each element as input in a new line will result in negative marking.)

   Output Format:

   - An integer n representing the most frequent neighborhood ID.
   - Each order number with it's respective neighborhood ID and OTP in a new line.

   NOTE: The OTPs can differ from the testcases since memory address will not be the same for every system. The OTPs are subject to change after each compilation of your code too.

   Testcases:

   - **Input:**

     ```
     5
     101 202 101 303 101
     ```

     Output:

```
:: Blinkit Delivery Service Coverage Report ::
###################################################
Most frequently visited neighbourhood : 101
###################################################
Additional Logs:
-- Order no.: 0, Nbh. ID: 101, OTP: 0x55973b58c140
-- Order no.: 1, Nbh. ID: 202, OTP: 0x55973b58c144
-- Order no.: 2, Nbh. ID: 101, OTP: 0x55973b58c148
-- Order no.: 3, Nbh. ID: 303, OTP: 0x55973b58c14c
-- Order no.: 4, Nbh. ID: 101, OTP: 0x55973b58c150
```

- **Input:**

```
6
50 100 50 100 150 50
```

Output:

```
:: Blinkit Delivery Service Coverage Report ::
###################################################
Most frequently visited neighbourhood : 50
###################################################
Additional Logs:
-- Order no.: 0, Nbh. ID: 50, OTP: 0x55973b58c140
-- Order no.: 1, Nbh. ID: 100, OTP: 0x55973b58c144
-- Order no.: 2, Nbh. ID: 50, OTP: 0x55973b58c148
-- Order no.: 3, Nbh. ID: 100, OTP: 0x55973b58c14c
-- Order no.: 4, Nbh. ID: 150, OTP: 0x55973b58c150
-- Order no.: 5, Nbh. ID: 50, OTP: 0x55973b58c154
```

- **Input:**

```
1
420
```

Output:

```
:: Blinkit Delivery Service Coverage Report ::
###################################################
Most frequently visited neighbourhood : 420
###################################################
Additional Logs:
-- Order no.: 0, Nbh. ID: 420, OTP: 0x55973b58c140
```

(6 marks)

All the Best!