import java.util.\*;

public class PageReplacement {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

while (true) {

// Accept frame size and reference string from user

System.out.print("Enter the number of frames: ");

int frameSize = scanner.nextInt();

System.out.print("Enter the number of pages: ");

int n = scanner.nextInt();

int[] referenceString = new int[n];

System.out.print("Enter the reference string (space-separated): ");

for (int i = 0; i < n; i++) {

referenceString[i] = scanner.nextInt();

}

// Select algorithm(s)

System.out.println("\nSelect the page replacement algorithm(s) (space-separated):");

System.out.println("1. FIFO");

System.out.println("2. LRU");

System.out.println("3. Exit");

System.out.print("Enter choices (e.g., 1 2 or 3 to exit): ");

String input = scanner.next(); // To read the entire line of input

String[] choices = input.split("\\s+"); // Split the input based on space

boolean exit = false;

for (String choice : choices) {

switch (choice) {

case "1":

fifoPageReplacement(referenceString, frameSize);

break;

case "2":

lruPageReplacement(referenceString, frameSize);

break;

case "3":

System.out.println("Exiting the program...");

exit = true;

break;

default:

System.out.println("Invalid choice. Please select again.");

break;

}

if (exit) {

scanner.close();

return; // Exit the program if the user selects option 3

}

}

// Ask again for more algorithms or to exit

System.out.println(); // Print a blank line for better readability

}

}

// FIFO Page Replacement Algorithm

private static void fifoPageReplacement(int[] referenceString, int frameSize) {

Set<Integer> frames = new LinkedHashSet<>();

Queue<Integer> fifoQueue = new LinkedList<>();

int pageFaults = 0;

for (int page : referenceString) {

if (!frames.contains(page)) {

if (frames.size() == frameSize) {

int oldestPage = fifoQueue.poll();

frames.remove(oldestPage);

}

frames.add(page);

fifoQueue.add(page);

pageFaults++;

}

}

System.out.println("Total page faults (FIFO): " + pageFaults);

}

// LRU Page Replacement Algorithm

private static void lruPageReplacement(int[] referenceString, int frameSize) {

Set<Integer> frames = new LinkedHashSet<>();

LinkedList<Integer> lruList = new LinkedList<>();

int pageFaults = 0;

for (int page : referenceString) {

if (!frames.contains(page)) {

if (frames.size() == frameSize) {

int lruPage = lruList.removeFirst();

frames.remove(lruPage);

}

frames.add(page);

lruList.add(page);

pageFaults++;

} else {

lruList.remove((Integer) page);

lruList.add(page);

}

}

System.out.println("Total page faults (LRU): " + pageFaults);

}

}