

Implement **question no:2** in the following list of questions during today's (7<sup>th</sup> January, 2016) lab session and upload your C program in the eduserver **before 4 pm**.

Naming Conventions for submission:

Submit a single C (.c) file (do not submit in any other archived formats like .rar or .tar.gz). The name of this file must be FIBSEQ\_<ROLLNO>\_<FIRST-NAME>.c (For example: FIBSEQ\_BxyyyyCS\_LAXMAN.c). DO NOT add any other files (like temporary files, input files, etc.) except your source code for submission.

**Questions:**

1. Write a C program that takes two 2-dimensional matrices as input, and prints their product, if they are compatible.

Input: Matrices  $A_{p \times q}$  and  $B_{m \times n}$  where  $1 \leq p, q, m, n \leq 750$

Output: Product of A and B, if they are compatible, "Incompatible matrices" otherwise.

2. Given an integer **num**, output whether **num** is present in the Fibonacci sequence or not. The Fibonacci sequence,  $F_n$  is defined by the recurrence relation  $F_n = F_{n-1} + F_{n-2}$ , with  $F_0 = 0$ ,  $F_1 = 1$  and  $F_2 = 1$ .

Input: An integer **num** in the range 0 to  $2^{31}$

Output: YES, if **num** is in the Fibonacci sequence, NO otherwise.

3. Given two strings, say  $s_1$  and  $s_2$ , write a program that outputs a string  $s_3$ , as follows: Characters of  $s_1$  and  $s_2$  occupy odd and even numbered positions in  $s_3$ , respectively. If either of  $s_1$  or  $s_2$  length is shorter than the other, then assign all characters in the shorter string to the respective positions of  $s_3$ . Also, left out characters in the longer string are appended directly at the end of  $s_3$ .

Input: Two strings  $s_1$  and  $s_2$  of length in the range 0 -  $2^{30}$

Output: String  $s_3$  with the characters from the input strings  $s_1$  and  $s_2$  interleaved.

Example:

Input:

Enter string 1:

Alice

Enter string 2:

boB

Output:

AbloiBce

4. Given two positive integers **n** and **k**, output all permutations of size **k** of the first **n** natural numbers.

Input: Positive integers **n** and **k**, with  $1 < k \leq n$ .

Output: All permutations of size **k** of the numbers **1** to **n**.

Example:

Input:

Enter value for **n**:

3

Enter value for **k**:

2

Output:

1 2

2 1

1 3

3 1

3 2

2 3

\*\*\*\*\*