CS110: Computer Programming Lab
Department of CSE
IIT, Guwahati

Module 04 Stage 02 Exercise 11

Assessment exercises are designed to help us check if the student has learned the basics of the topics included in the drill instructions. However, a drill assessment is not a comprehensive assessment. A fuller and complete assessment aimed at determining the course grades will be done through CS110 examinations.

It is expected that the student will attempt and solve many more exercises from the drill assessment sets to improve their programming skills and for an excellent performance at the examinations.

Exercise

Consider the following list of students with their roll numbers and gender information arriving for a dance lesson. Each dancing pair has two persons of different gender. The students are listed in order of their arrival. Write a program to pair the students for the dance lesson based on the order they arrive. That is, first male and first female will form the first pair for the lesson.

```
1000 Ram Male
1002 Radha Female
1004 Alka Female
1010 Mohan Male
1018 Rahim Male
1007 Narendra Male
1005 Soniya Female
1200 Rahul Male
1150 Jyoti Female
1127 Sita Female
1146 Krishna Male
1176 Padmavati Female
0 Ignored Ignored
```

Print the list of pairs for the lessons by listing names for each pair in a line.

Two ways to do this program

You must write this program in two different ways.

Record all Arrivals before Pairing

Define two queues; one for males and other for females. Include each arrival into a queue. When the arrivals cease, form pairs and list the pairs on screen.

Form Pairs on the Fly

In this arrangement, there is only one queue. And there is a separate variable used to record the gender of entries in the queue. All entries in the queue are of same gender.

Pairs are formed as soon as they can be formed. Print the pairs when formed without waiting for the last arrival to be marked.

Interface queue

```
#ifndef QUEUE_H_INCLUDED
#define QUEUE_H_INCLUDED

/* Returns a reference (pointer) to a new queue */
void * mkQueue(void);

/* Removes a queue specified by a valid reference */
int rmQueue (void *);

/* Returns number of entries in a validly referenced queue */
int sizeQ (void *);

/* Place objP  in queueP and returns new count of entries */
int joinQ (void * queueP, void * objP);

/* Return earliest arrived objP in queue */
void * leaveQ (void *);

#endif // QUEUE_H_INCLUDED
```