

## 1. 2016-1

Circle or cross: "T" if True – "F" if False.

- T / F A semaphore is a data structure.
- T / F Semaphores can not be used for avoiding dead locks
- T / F A monitor is a programming language construct
- T / F Monitors encapsulate shared data structures.
- T / F Both semaphores and monitors are distributed as function calls.
- T / F Monitors use condition variables, while semaphores do not.

## 2. 2016-2

```
001 /*                                030 void* thread3 (void* a) {
002  * (c) 2015-2016 Rahmat M. Samik-Ibrahim 031     printf("T3X\n");
002  * -- This is free software              032     sem_post (&sem[6]);
003  * Feel free to copy and/or modify and/ 033     sem_post (&sem[2]);
004  * or distribute it, provided this notice, 034 }
004  * and the copyright notice, are preserved. 035
005  * REV04 Tue Dec 13 15:19:04 WIB 2016     036 void* thread4 (void* a) {
006  * START Wed Sep 30 00:00:00 UTC 2015     037     sem_wait (&sem[4]);
007  */                                       038     printf("T44\n");
008                                           039     sem_wait (&sem[5]);
009 #include <stdio.h>                               040     printf("T45\n");
010 #include <stdlib.h>                               041     sem_wait (&sem[6]);
011 #include <semaphore.h>                           042     printf("T46\n");
012 #include "99-myutils.h"                         043 }
013 #define nSem 7                                   044
014                                           045 void main(void) {
015 sem_t sem[nSem];                                046     printf("MAIN\n");
016                                           047     for (int ii=1;ii<nSem;ii++)
017 void* thread1 (void* a) {                       048         sem_init(&sem[ii], 0, 0);
018     sem_wait (&sem[1]);                          049     daftar_trit (thread1);
019     printf("T1X\n");                               050     daftar_trit (thread2);
020     sem_post (&sem[4]);                          051     daftar_trit (thread3);
021 }                                               052     daftar_trit (thread4);
022                                           053     jalankan_trit ();
023 void* thread2 (void* a) {                       054     beberes_trit ("TREXIT");
024     sem_wait (&sem[2]);                          055 }
025     printf("T2X\n");
026     sem_post (&sem[5]);
027     sem_post (&sem[1]);
028 }
```

Write down the program output: