# מכללה אקדמית הדסה
## החוג למדעי המחשב

## תרגיל 5#: תכנות מערכת ומבוא לתכנות מקבילי-- זיכרון משותף

**תכנית a: מילוי מערך – בעזרת ז"מ**
נחזור למשימה המוכרת של תהליך אחד שהינו בעל מערך, ותהליכים אחרים המייצרים מספרים ראשוניים שיוספו ממוינים למערך. הפעם הכלי לתקשורת בין תהליכים בו נשתמש יהיה הז"מ.

תכנית א': בעל המערך
מגדירה בזיכרון משותף מערך של מספרים שלמים בן ARR_SIZE (קבעוהו להיות 105) תאים.
לתא הראשון בז"מ מכניס התהליך את מספרו,
לשלושת התאים הבאים הוא מכניס אפס (שיוסבר מיד),
גם לתא החמישי שישמש כמעין 'מנעול' פרימיטיבי, הוא מכניס אפס (הסבר על כך בהמשך).
את יתר התאים (100) הוא מאפס.
עתה הוא הולך לשון עד קבלת סיגנל.
עת הוא מקבל סיגנל הוא מתעורר.
הוא מציג כמה ערכים יש במערך, את הערך המזערי והמרבי.
הוא משחרר את הזיכרון, ומסיים.

זמנו את ftok("." ,'5')

תכנית ב' (ממנה תריצו שלושה עותקים): יצרני מספרים
תכנית זאת תתחבר לז"מ שהקצתה תכנית א'. התכנית תתנהל בלולאה. בכל סיבוב בלולאה התכנית מייצרת מספר במידה והוא ראשוני וגדול או שווה מהערך הגדול ביותר המצוי במערך היא מוסיפה אותו למערך שבז"מ. היא סופרת כמה ערכים חדשים היא הוסיפה. במידה ומאה פעמים ברצף היא נכשלה בהוספת מספר ראשוני היא מתנהלת כמו במצב בו המערך מלא.

כדי לוודא ששני יצרנים לא יוסיפו ערך במקביל, וכך, אולי ידרסו אחד את הערך שהוסיף משנהו, היצרן, טרם שהוא מוסיף מספר ראשוני, פונה לתא החמישי (המנעול) כל עוד ערכו 0 (כלומר הוא 'נעול') הוא ממתין. עת ערכו הוא אחד (הוא 'פתוח') הוא משנה את ערכו לאפס (נועל אותו), ובכך זוכה בזכות להוסיף ערך למערך לבדו. לבסוף הוא 'פותח' את המנעול: מכניס לו את הערך 1.

אם היצרן מצא שכל תאי המערך מלאים (או הוא נכשל מאה פעמים ברצף בהוספת מספר ראשוני) הוא מציג את אותו פלט כמו בתרגיל הקודם (כמה ערכים הוא הוסיף למערך), הוא שולח סיגנל לבעל המערך, ואז הוא מסיים.

שלושת תאי המערך הראשונים ישמשו אותנו כדי לגרום ליצרנים להתחיל למלא את המערך (פחות או יותר) ביחד: כל יצרן עת מורץ מכניס 'לתא שלו' מבין השלושה את הערך 1. רק אחרי שהוא מגלה שבכל שלושת התאים של שלושת היצרנים מצוי הערך 1, הוא נכנס ללולאה המרכזית של הוספת הערכים.

כמו בתרגיל הקודם, נקבע שהיצרנים יקבלו דרך וקטור הארגומנטים את המספר שלהם, שיהיה גם ה: seed שלהם: 1, 2, 3.

הערה: הסבירו בקובץ ה: README אילו מצבי מרוץ קיימים בתכנית.

**תכנית b: שרת ראשוניות ושרת פלינדרומיות – חזרה על תכנית c מתרגלי #4**

כתבו את שלוש התכניות הבאות:

שרת ראשוניות

עם תחילת ריצתו מקצה ז"מ בן 102 תאים.
לתוך התא 0# הוא מכניס את ה pid שלו.
לתא 1# לקוח שלו יכניס את ה: pid של הלקוח
את יתר מאה תאי המערך הוא מאפס. לתאים אלה לקוח שלו יכניס לכל היותר מאה מספרים שאת ראשוניותם יש לבדוק (ולאחריהם אפס שיציין את סוף הסדרה).

השרת רץ בלולאה אינסופית. בכל סיבוב בלולאה הוא:
(א) הולך לשון עד קבלת הסיגנל SIGUSR1.
(ב) הוא עובר על תאי המערך בהם קיים ערך חיובי ממש ובכל אחד מהם שמכיל ערך פריק (לא ראשוני) הוא שם 1-. הוא שולח את הסיגנל SIGUSR1 ללקוח שפנה אליו

שרת פלינדרומיות.

מתנהל באופן דומה. נקבע שתאי הז"מ שלו יהיו:
0# = ה: pid שלו.
1# = ה: pid של הלקוח
2# עד 20# = סדרת המספרים שאת הפלינדרומיות שלה יש לבדוק (כלומר האם היא סימטרית או לא)
21# = תוצאת הבדיקה.
שרת זה יעורר באמצעות הסיגנל SIGUSR2, וזה יהיה גם הסיגנל שהוא יישלח.

תהליך front end

בלולאה אינסופית:
(א) קורא מהמשתמש את הערך p ואחריו סדרת מספרים שיש לבדוק את ראשוניותם עד קריאת הערך אפס שמסמן את סוף הסדרה (וניתן להניח שאורך הסדרה יהיה < 100), או את הערך q ואחריו סדרת מספרים הנגמרת באפס ושאת הפלינדרומיות שלה יש לבדוק (לא כולל האפס). ניתן להניח שהסדרה לא ארוכה מדי.
(ב) מאחסן את הקלט בז"מ הדרוש.
(ג) שולח סיגנל המתאים לשרת הדרוש.
(ד) הולך לשון עד קבלת הסיגנל SIGUSR2/SIGUSR1.
(ה) עת מתעורר, שולף את הפלט, ומציג אותו למשתמש.

כל תכנית תסתיים ע"י קבלת הסיגנל SIGINT. ב: signal handler ישוחררו קטעי הז"מ שהוקצו.

נקבע של: ftok נעביר את ".", וכן את התווים: p, q.

שמות הקבצים לפי הקונבנציות שקבענו בתחילת הסמ':
ex5b1.c = שרת ראשוניות.
ex5b2.c = שרת פלינדרומיות
front end = ex5b3.c

שאלה: האם השרתים הללו יוכלו לקבל פניה מכמה תהליכי front end שירוצו במקביל?
במידה ולא: מה ניתן לשנות בהם ע"מ שהדבר יתאפשר?