# מטלה - החלפת בתים וכליות

א. ממשו את אלגוריתם מעגלי המסחר בשפת-תיכנות לבחירתכם או בפסאודו-קוד. היעזרו במחלקות הבאות (ניתן להוסיף להן שדות לפי הצורך):

```
struct Worker{
      string name; // worker name, for display
      vector<Shift*> preferences;
            // preferences[0] is the best shift for the worker.
            // preferences[1] is the 2nd-best shift for the worker. etc...
      Shift* current_shift;
            // The shift to which the worker is currently assigned
};
struct Shift {
      string name; // shift name, for display
      Worker* current_worker;
            // The worker to which the shift is currently assigned
};
```

הפונקציה מקבלת כקלט וקטור עם כל העובדים במערכת (לכל עובד רשום סדר העדיפויות שלו והמשמרת הנוכחית שלו), ווקטור עם כל המשמרות במערכת (לכל משמרת רשום מי העובד הנוכחי המשובץ לאותה משמרת). הפונקציה אמורה לכתוב למסך את השינויים שיש לבצע בלוח השיבוצים, למשל:

Avraham moves from morning shift to afternoon shift.
Isaac moves from afternoon shift to morning shift.

כותרת הפונקציה:

```
void replace(vector<Worker*> workers, vector<Shift*> shifts);
```

ב. הדגימו את פעולת הפונקציה על הקלט הבא. הסבירו מה תדפיס הפונקציה שלכם ומדוע.

```
workers = [
      {name: Avraham,
       preferences: [morning, noon, evening],
       current_shift: noon},
      {name: Isaac,
       preferences: [noon, evening, morning],
       current_shift: evening},
      {name: Yaacov,
       preferences: [evening, morning, noon],
       current_shift: morning}
]
shifts = [
      {name: morning, current_worker: Yaacov},
      {name: noon, current_worker: Avraham},
      {name: evening, current_worker: Isaac},
]
```

א. פתרון: לצורך מציאת המעגל הוספתי שדה בוליאני visited ל-Worker.

```
void replace(vector<Worker*> workers, vector<Shift*> shifts) {
      while (workers.size() > 0) {  // while there are workers in the system:
            // find a cycle in the top-choice graph:
            for (auto w: workers)
                  w→visited = false;
            w = workers[0];
            w→visited = true;
            do {
                  w = w→preferences[0]→current_worker;
                  if (w→visited) { // found a cycle:
                        // exchange in the cycle, and remove from lists:
                        auto first_in_cycle = w;
                        do {
                              auto w_new_shift = w→preferences[0];
                              cout << w→name << " moves from "
                                   << w→current_shift << " to "
                                   << w_new_shift;
                              workers.remove(w);
                              for (auto w2 in workers) {
                                    w2.preferences.remove(w_new_shift);
                              }
                        } while (w!=first_in_cycle);
                  } // end of exchange in cycle
            } // end of looking for cycle (we must find a cycle!)
      }
}
```

ב. פתרון:

בהתחלה w שווה לעובד Avraham. ואז הולכים לעובד הנוכחי של משמרת morning ומגיעים ל-Yaacov. הולכים לעובד הנוכחי של משמרת evening ומגיעים ל Isaac. הולכים לעובד הנוכחי של משמרת noon ומגיעים לאברהם. עכשיו visited של אברהם הוא true ולכן מצאנו מעגל. בתוך הלולאה הפנימית, נעבור על המעגל החל מאברהם (שהוא הראשון במעגל), ונדפיס:

```
Avraham moves from noon to morning
Yaacov moves from morning to evening
Isaac moves from evening to noon
```

# מאמרים להרחבה ולמטלת-רשות


1. C Hajaj, JP Dickerson, A Hassidim, T Sandholm (2015): "Strategy-Proof and Efficient Kidney Exchange Using a Credit Mechanism"
2. I Ashlagi, F Fischer, IA Kash, AD Procaccia (2015): "Mix and match: A strategyproof mechanism for multi-hospital kidney exchange".
3. DJ Abraham, A Blum, T Sandholm (2007): "Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges".
4. MU Ünver (2010): "Dynamic kidney exchange"
5. JP Dickerson, AD Procaccia, T Sandholm (2013): "Failure-aware kidney exchange"
6. JP Dickerson, AD Procaccia, T Sandholm (2014): "Price of fairness in kidney exchange"
7. I Ashlagi, AE Roth (2012): "New challenges in multihospital kidney exchange"
8. S Luo, P Tang (2015), "Mechanism design and implementation for lung exchange"
9.