

# נושאים מתקדמים בתכנות מונחה עצמים: תרגיל מעבדה 19/04/2023

## תרגיל 1

בהינתן הפונקציה הבאה,

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 template<class IterIn, class IterOut>
5 void keep_it_real(IterIn first, IterIn last, IterOut out) {
6     std::vector< std::pair<IterIn,int> > s(last-first);
7     for(int i=0; i < s.size(); ++i)
8         s[i] = std::make_pair(first+i, i );
9
10    /* TODO 1: std::sort s according to the value of the first
11       element within the pair */
12    /* TODO 2: copy s into the IterOut instance */
13 }
```

ממשו את קטעי הקוד החסרים על מנת להשיג את הפונקציונליות הבאה:

1. מיון הווקטור  $s$  בשורות 10-11, בעזרת `std::sort` ושימוש בתחביר  $\lambda$ , בהסתמך על קריטריון המיון הבא - יש למיין ע"פ הערך של האיבר הראשון בצמדים המוחזקים בווקטור.
2. העתקת הערכים בווקטור  $s$  לאחר המיון אל המופע של `IterOut` בשורה 12.
3. הקפידו על כתיבת לולאות `for` של התקן החדש.

## תרגיל 2

בהינתן הפונקטור הבא,

```
class Modulo_print {
    ostream& os;
    int m;
public:
    Modulo_print(ostream& s, int mm) :os(s), m(mm) {}
    void operator()(int x) const
        { if (x%m==0) os << x << '\n'; }
};
```

נתונה הפונקציה הגלובלית –

```
void print_modulo(const vector<int>& v, ostream& os, int m)
    // output v[i] to os if v[i]%m==0
{
    for_each(begin(v), end(v), Modulo_print{os,m});
}
```

מטרתכם היא לשכתב קוד זה באופן גנרי, בתחביר מודרני וקומפקטי, וליצור שתי גרסאות של `print_modulo` (תוך ביטול מחלקת `Modulo_print`):

1. מימוש באמצעות מהלך *depth-first* על הקונטיינר
  2. מימוש באמצעות מהלך *breadth-first* על הקונטיינר
- צינו כל הנחה שאתם מבצעים בכתיבת הפיתרון.

בהצלחה רבה!