

# תכנות מתקדם ושפת C++

## תרגיל 2

### אלגוריתמים

מועד אחרון להגשה: 21.12.2019

בתרגיל נממש אלגוריתמים שלא נמצאים בספריה הסטנדרטית.

הנחיות הגשה:

- ההגשה דרך מערכת ההגשה בכתובת:

<http://submit.org.il>

- יש לממש שלושה אלגוריתמים:
  1. Median
  2. Transpose
  3. Transformations
- המימוש של שלושת האלגוריתמים יהיה כלול בתוך קובץ אחד בשם `alg.h`
- יש לכתוב קובץ `main.cpp` שיעשה `include` לקובץ `alg.h` ויבדוק את האלגוריתמים.
- להגיש רק את הקובץ: `alg.h`

תאור האלגוריתמים:

#### 1. Median

`p = Median(b, e, n)`

האלגוריתם מקבל איטרטור `b` לתחילת מיכל ואיטרטור `e` לסוף המיכל ופרמטר שלישי `n` שמציין את מספר האברים.

האלגוריתם מוצא את החציונים של כל `n` איברים ושם אותם בתחילת המיכל.

אם המיכל ריק או שהערך של `n` הוא 1 או שמספר האיברים לא מתחלק בשלמות ב-`n`, האלגוריתם לא יחשב את החציונים והערך המוחזר יהיה האיטרטור `e`.

אחרת, האלגוריתם יחשב את החציונים והערך המוחזר יהיה איטרטור לסוף (אחד אחרי) החציונים.

דוגמה:

```
int ia[] = {1,2,3,10,8,6}
int * p = Median(begin(ia), end(ia), 3);
```

יחזיר:

```
{2,8,↓, , , }
```

p יצביע על האיבר שאחרי 8.

## 2. Transpose

```
P = Transpose(b,e)
```

האלגוריתם מקבל איטרטור b לתחילת מיכל ואיטרטור e לסוף המיכל.

האלגוריתם מחליף בין שני איברים סמוכים, בין איבר 0 לאיבר 1, בין איבר 2 לאיבר 3 וכן הלאה.

אם מספר האיברים אינו זוגי, האיבר האחרון לא יוחלף.

האלגוריתם מחזיר את האיטרטור e אם מספר האיברים זוגי אחרת מחזיר e - 1.

לצורך ההחלפה אפשר להשתמש באלגוריתם iter\_swap.

דוגמה:

```
int ia[] = {1,2,3,10,8,6}
int * p = Transpose(begin(ia), end(ia));
```

יחזיר:

```
{2,1,10,3,6,8}
```

## 3. Transformations

```
p = Transformations(b,e,f)
```

האלגוריתם מקבל איטרטור b לתחילת מיכל, איטרטור e לסוף המיכל ופרמטר שלישי פונקציה.

האלגוריתם מפעיל את הפונקציה על כל שני איברים סמוכים: איבר 0 ואיבר 1, איבר 2 ואיבר 3 וכן הלאה, ושם את התוצאות בתחילת המיכל.

אם המיכל ריק או מספר האיברים אינו זוגי, האלגוריתם לא יחשב את הפונקציה והערך המוחזר יהיה האיטרטור e.

אחרת, הערך המוחזר יהיה איטרטור לסוף (אחד אחרי) התוצאות.

דוגמה:

```
int ia[] = {1,2,3,10,8,6}
int * p = Transformations(begin(ia), end(ia),
    [](const int x, const int y){return x + y;});
```

יחזיר:

```
{3,13,14,↓, , }
```

**הערות:**

יש לממש את האלגוריתמים עם templates כך שיוכלו לפעול עבור מיכלים שונים.

אפשר למצוא ברשת את המימוש של האלגוריתמים הסטנדרטיים ולהתאים אותם.

## בהצלחה