

תכנות מתקדם ושפת C++

תרגיל 2

אלגוריתמים

מועד אחרון להגשה: 15.5.2020

בתרגיל זה נממש אלגוריתמים שלא נמצאים בספריה הסטנדרטית.

- ההגשה ביחיד לכתובת:

<http://submit.org.il/ariel>

- צריך לממש שלושה אלגוריתמים:

1. `Add_n`

2. `Transpose`

3. `Transform`

- צריך לכתוב קובץ כותרת בשם `alg.h` שיכיל מימוש של שלושת האלגוריתמים.

- צריך להגיש רק את הקובץ: `alg.h`

- צריך לממש את האלגוריתמים עם `templates` כך שיוכלו לפעול עבור מיכלים עם איברים שונים.

- אפשר למצוא ברשת את המימוש של האלגוריתמים הסטנדרטיים ולהתאים אותם.

בהצלחה

להלן תיאור האלגוריתמים:

1. Add_n

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

האלגוריתם מקבל איטרטור **b** לתחילת מיכל, איטרטור **e** לסוף מיכל ומספר חיובי **n**.

האלגוריתם מחבר את **n** האיברים הראשונים ושומר את הסכום במקום הראשון במיכל, מחבר את **n** האיברים הבאים ושומר את הסכום במקום השני במיכל, וכן הלאה.

הערך המוחזר הוא איטרטור לסוף (אחד אחרי) הסכומים.

אם המיכל ריק הערך המוחזר הוא **e**.

אם מספר האיברים לא מתחלק בשלמות ב- **n**, הסכום האחרון יהיה החיבור של (1 עד **n-1**) האיברים הנותרים.

דוגמה 1:

```
int ia[] = {1,2,3,4,5,6,7,8,9};  
int * p = Add_n(begin(ia), end(ia), 3);
```

תוצאה:

```
{6,15,24,4,5,6,7,8,9}
```



p יצביע על האיבר שאחרי 24.

דוגמה 2:

```
int ia[] = {1,2,3,4,5,6,7,8,9};  
int * p = Add_n(begin(ia), end(ia), 2);
```

תוצאה:

```
{3,7,11,15,9,6,7,8,9}
```



p יצביע על האיבר שאחרי 9.

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.

אם מספר האיברים זוגי או שהמיכל ריק, הערך המוחזר הוא **e**.

אם מספר האיברים אינו זוגי, הערך המוחזר הוא **e - 1**.

לצורך ההחלפה אפשר להשתמש באלגוריתם **iter_swap**.

דוגמה:

```
int ia[] = {1,2,3,4,5,6,7,8,9};
```

```
int * p = Transpose(begin(ia), end(ia));
```

תוצאה:

```
{2,1,4,3,6,5,8,7,9}
```



3. Transform

```
p = Transform(b,e,o,f)
```

האלגוריתם מקבל איטרטור **b** לתחילת מיכל, איטרטור **e** לסוף מיכל, איטרטור שלישי **o** לתחילת מיכל אחר ופונקציה **f** שמקבלת שני פרמטרים.

האלגוריתם מפעיל את הפונקציה **f** על שני האיברים הראשונים במיכל הראשון (איבר 0 ואיבר 1) ושומר את התוצאה באיבר הראשון של המיכל השני, האלגוריתם מפעיל את הפונקציה על שני האיברים הבאים במיכל הראשון (איבר 2 ואיבר 3) ושומר את התוצאה באיבר השני של המיכל השני, וכן הלאה.

אם מספר האיברים זוגי הערך המוחזר הוא איטרטור למקום אחד אחרי האיבר האחרון במיכל השני. אם המיכל ריק או מספר האיברים אינו זוגי, האלגוריתם לא יחשב את הפונקציה והערך המוחזר יהיה איטרטור לתחילת המיכל השני.

דוגמה:

המיכל הראשון:

```
int ia[] = {1,2,3,10,8,6};
```

המיכל השני:

```
vector<int> vec(100);
```

```
int * p = Transform(begin(ia), end(ia), vec.begin(),  
    [](const int x, const int y){return x + y;});
```

יחזיר:

```
{3,13,14, , , // vec  
  ↑
```