

# Itertools

## Functions creating iterators for efficient looping

BY: Nathan almaleh 209633635 && Adam Lev Ari 204115547

במטלה זו נלמד לבנות מבנים "דמויי-מיכלים", שאפשר לרוץ עליהם בלולאה למרות שאינם תופסים מקום בזיכרון. השם הרשמי של מבנה מסוג זה הוא

## Iterable

נבנה חמישה דמויי מיכלים כאלו.

### 1. Cycle

```
vector<int> vecInit = {1,2,3,4};  
cout << "#### Cycle: ####";  
cout << endl << "Finite<5> Cycle of numbers vector: " << endl;  
for (auto i: Cycle<vector<int>>(vecInit,5))  
    cout << i;    // 1234 1234 1234 ...  
cout << endl;
```

מייצג סדרה של עצמים , לדוגמה הוקטור, ועלו צריך לעשות איטרציות הדפסה אינסופית

### 2. Accumulate

```
vector<int> vecInit = {1,2,3,4};  
for (auto i: Accumulate<vector<int>>(vecInit,5))  
    cout << i;    // 1 3 6 10  
cout << endl;
```

מייצג סדרה של עצמים , לדוגמה הוקטור, ועלו צריך לעשות איטרציות הסוכמת את האיברים אבור מספרים וגם כן עבור אותיות .

### 3.zip\_longest

```
vector<String> vecInit = {A,B,C,D};  
vector<String> vecStr = {X,Y};  
for (auto i: Zip_Longest<vector<string>>(vecInit,vecStr))  
    cout << i;  
zip_longest('ABCD', 'xy', fillvalue='-') --> Ax By C- D-  
cout << endl;
```

מייצג סדרה של שני עצמים, לדוגמה הוקטורים, ועלו צריך לעשות איטרציות הסוכמת את האיברים עבור המיקום שלהם מספרים וגם כן עבור אותיות.

## 4.Filterfalse

```
vector<Int> vecInit = {0,1,2,3,4,5,6,7,8,9,10};
for (auto i: filterfalse <vector<Int>>(vecInit,Function))
    cout << i;
filterfalse(lambda x: x%2, range(10)) --> 0 2 4 6 8
cout << endl;
```

מייצג סדרה של עצמים עם פונקציה, לדוגמה הוקטורים והלמדה, ועלו צריך לעשות איטרציות הבדקת את האיברים ומדפיסה את הערכים שהם שלילי.

## 5.permutations

```
vector<String> vecInit = {ABC};
for (auto i: Permutation <vector<String>>(vecInit))
    cout << i;
all possible orderings, no repeated elements ABC ACB BAC BCA CBA CAB
cout << endl;
```

מייצג סדרה של עצמים, לדוגמה הוקטורים, ועלו צריך לעשות איטרציות המדפיסה את כל הפרמוטציות השונות.