

- ❖ في **Vector** يوجد **Method** اسمه **push_back** لإضافة العناصر
- ❖ عند إضافة عنصر ل **push_back(10)** يتم إضافتها في **Stack**
- ❖ **Vector** يستخدم نوع من أنواع **Data Structure** وهي **Stack**
- ❖ أول عنصر يدخل في **Stack** هو آخر عنصر يخرج من **Stack** (والعكس بالعكس)
- ❖ إضافة عنصر الى **Stack** يسمى **Push**
- ❖ إزالة عنصر من **Stack** يسمى **Pop**
- ❖ **Array** في **Parameters** هي **& By Reference**
- ❖ **Vector** في **Parameters** هي **By Value**
- ❖ عند إنشاء **Parameters** من نوع **Vector** يفضل دائما وضع إشارة **&**
- ❖ **By Reference** (للتعديل على **Vector** الرئيسي)
- ❖ يفضل دائما وضع إشارة **&** في **Ranged Loop** (للذهاب الى **Address** وعدم نسخه

```
#include <iostream>
#include <vector> // ديناميكية المخزنة القيم حسب على الذاكرة في مساحة لحجز
using namespace std;

int main()
{
    vector <int> vNumbers ;

    // إضافة عناصر الى vector وأخذ مساحة له بشكل أوتوماتيكي
    vNumbers.push_back(10);
    vNumbers.push_back(20);
    vNumbers.push_back(30);
    vNumbers.push_back(40);
    vNumbers.push_back(50);

    cout << "Numbers vector : \n\n";

    // الطرق لطباعة Vector هي Ranged Loop
    // int Number لا بد أن يكون مثل vector <int>
    // (int Number : vNumbers) ينسخ العنصر الأول من vector الى int Number ...
    // عملية النسخ تأخذ وقت وحجم = برنامج بطيء
    for (int & Number : vNumbers)
    {
        // (int & Number : vNumbers) إشارة & ، عملها الذهاب الى موقع العنصر في الذاكرة
        // إشارة & : الى عدم نسخ العناصر
        cout << Number << endl;
    }
    cout << "\n\n";
}
```