LECTURE 01

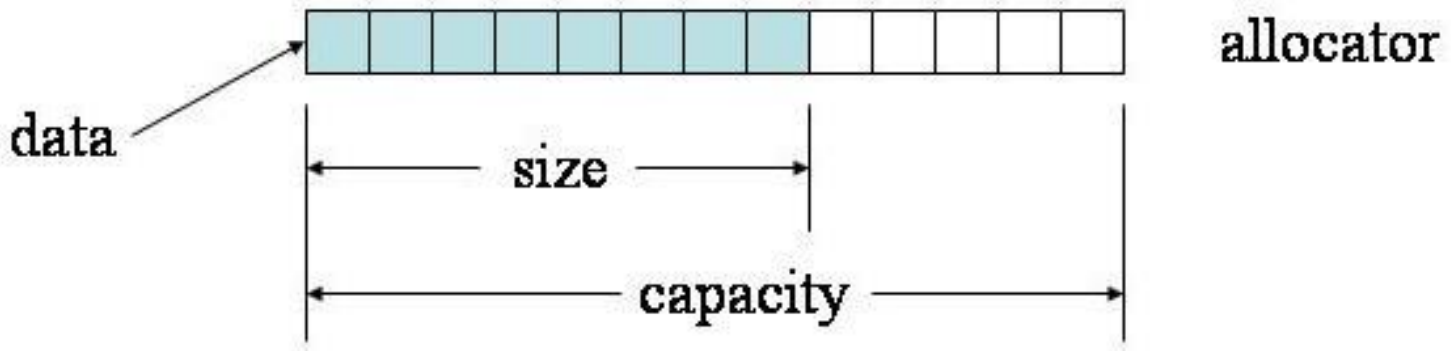
DYNAMIC ARRAY

Phạm Nguyễn Sơn Tùng

Email: sontungtn@gmail.com

Dynamic Array là gì?

Dynamic Array là một cấu trúc dữ liệu mảng động, người dùng không cần khai báo trước số lượng phần tử là bao nhiêu.



C++: `vector`

Java: `Arraylist`

Python: `list`

Cách khai báo sử dụng Dynamic Array

C++

Thư viện:

```
#include <vector>
using namespace std
```

Khai báo:

```
vector<data_type> name;
```

```
vector<int> v;
```

python

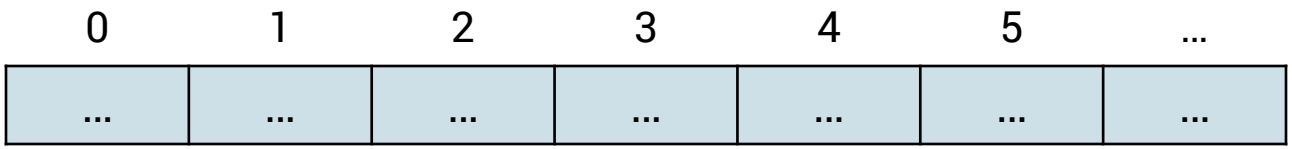
Thư viện:

```
NULL
```

Khai báo:

```
variable = [value1, value2, ..]
```

```
l = []
```



Cách khai báo sử dụng ArrayList

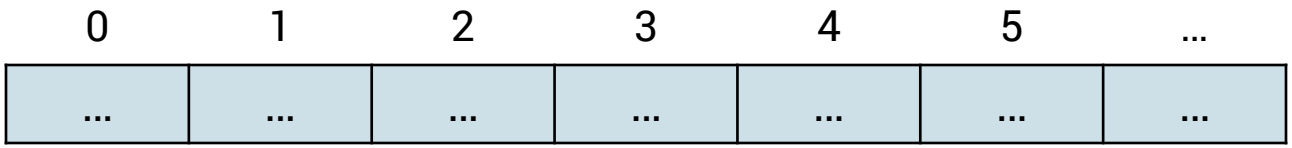
Thư viện:

```
import java.util.ArrayList;
```

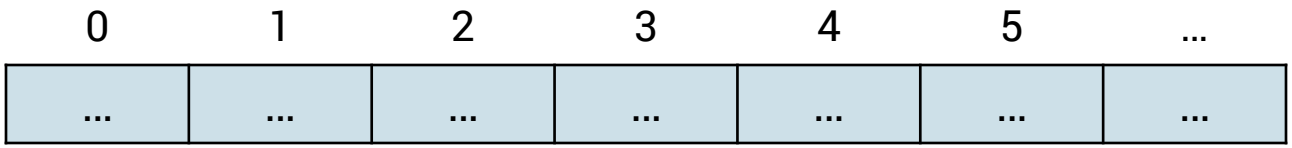
Khai báo:

```
ArrayList<DataType> variable = new ArrayList<DataType>();
```

```
ArrayList<String> a = new ArrayList<String>();
```



Thêm phần tử vào cuối Dynamic Array



C++

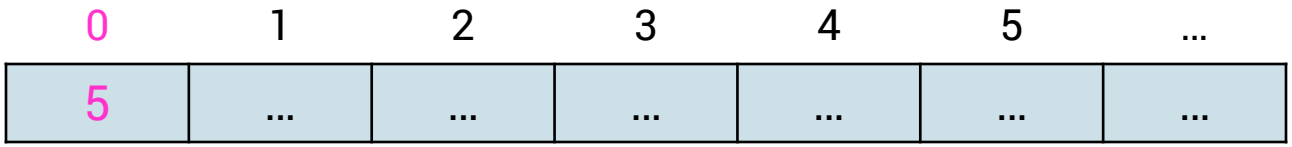
python

push_back(value)

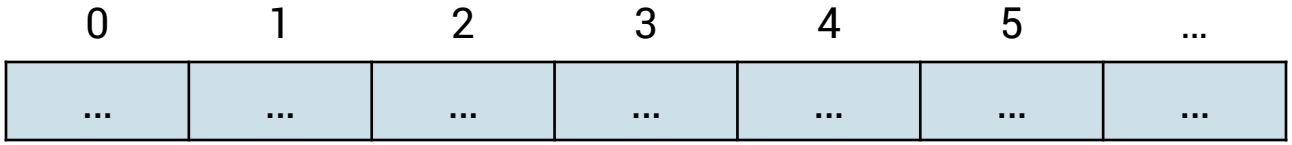
```
vector<int> v;
v.push_back(5);
```

append(obj)

```
l = []
l.append(5)
```

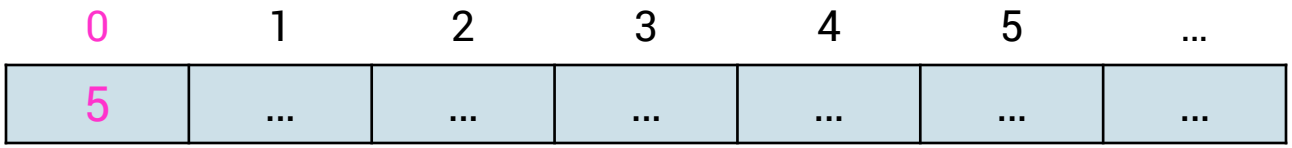


Thêm phần tử vào cuối ArrayList

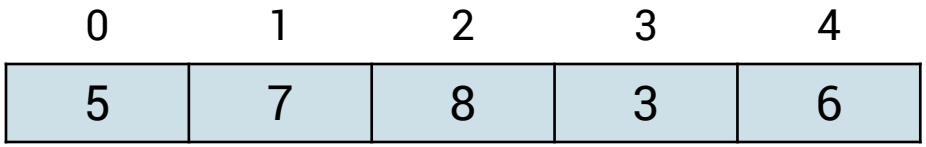


add(value)

```
ArrayList<Integer> a = new ArrayList<Integer>();
a.add(5);
```



Lấy phần tử đầu tiên Dynamic Array



C++

front()

```
int result = v.front();
//hoặc result = v[0];
cout<<result;
```

python

Dùng vị trí để trả về giá trị đầu tiên.

```
result = l[0]
print(result)
```

5

Lấy phần tử đầu tiên của ArrayList

0	1	2	3	4
5	7	8	3	6

get(index): Để lấy giá trị phần tử đầu hay cuối của ArrayList, ta sử dụng hàm get để lấy giá trị. Lưu ý chương trình sẽ Throw exception nếu index không thỏa mãn giới hạn của ArrayList (index < 0 hoặc index >= size()).

```
a.get(0);
```

5

Lưu ý: hàm get chỉ trả về giá trị, không thể dùng để cập nhật cho ArrayList.

Lấy phần tử cuối cùng Dynamic Array

0	1	2	3	4
5	7	8	3	6

C++

back()

```
int result = v.back();
// result = v.[n-1];
cout<<result;
```

python

Dùng vị trí để trả về giá trị cuối cùng.

```
result = l[-1]
print(result)
```

6

Lấy phần tử cuối cùng ArrayList

0	1	2	3	4
5	7	8	3	6

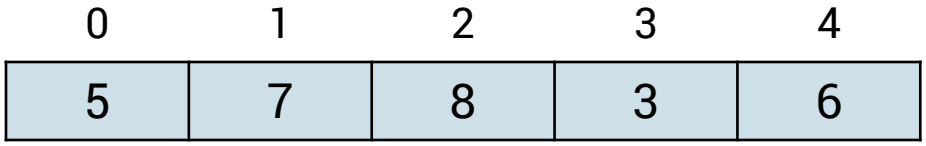
Tiếp tục dùng hàm **get(index)** để lấy phần tử cuối cùng.

```
a.get(a.size() - 1);
```

6

Lưu ý: hàm size lấy số lượng phần tử của ArrayList.

Chèn một giá trị vào Dynamic Array



C++

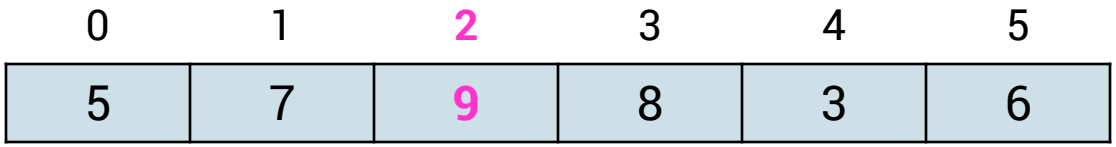
python™

insert(iterator, val): Chèn **một** giá trị.

```
vector<int>::iterator it;
it = v.begin() + 2;
v.insert(it, 9);
```

insert(pos, val): chèn giá trị val vào vị trí pos.

```
l.insert(2, 9)
```



Chèn một giá trị vào ArrayList

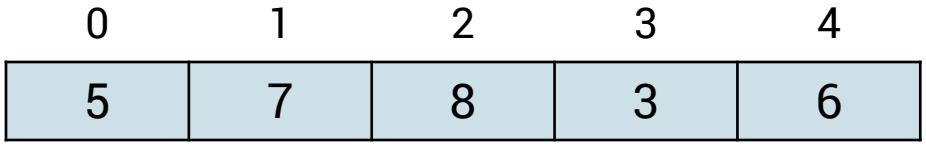
0	1	2	3	4
5	7	8	3	6

add(index, value): Sử dụng lại hàm add có thêm một tham số thứ 2 để chèn một phần tử vào **vị trí index** của ArrayList. Throw exception nếu index không thỏa mãn giới hạn của ArrayList (index < 0 hoặc index > size()).

```
a.add(2, 9);
```

0	1	2	3	4	5
5	7	9	8	3	6

Chèn hàng loạt giá trị vào Dynamic Array



C++

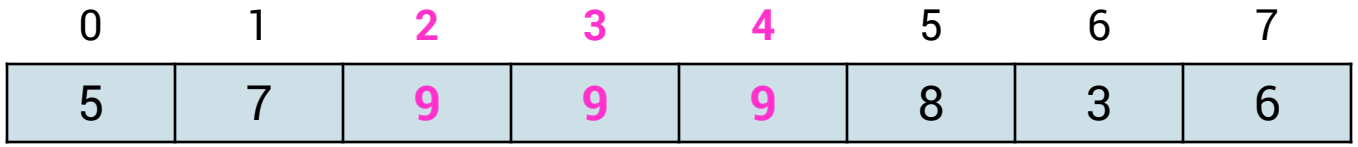
insert(iterator, size_type, value)

```
vector<int>::iterator it;
it = v.begin() + 2;
v.insert(it, 3, 9);
```

python™

<variable>[pos:pos] = n*[<value>]

```
l[2:2] = 3*[9]
```



Chèn hàng loạt vào ArrayList

Java không có hàm chèn nhiều phần tử cùng giá trị vào ArrayList.

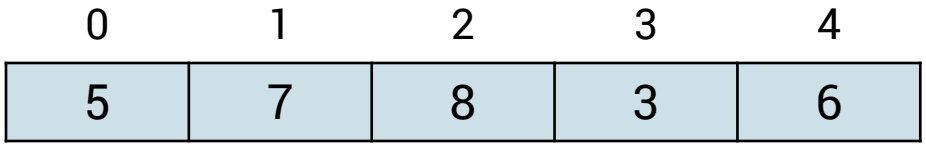
0	1	2	3	4
5	7	8	3	6

```
static void addMultipleValue(ArrayList<Integer> a, int pos, int num, int val) {
    if (pos < 0 || pos > a.size() || num <= 0)
        return;
    for (int i = 0; i < num; i++)
        a.add(val);
    for (int i = 0; i < a.size() - num - pos; i++)
        a.set(a.size() - i - 1, a.get(al.size() - num - i - 1));
    for (int i = 0; i < num; i++)
        a.set(pos + i, val);
}
```

```
addMultipleValue(a, 2, 3, 9)
```

0	1	2	3	4	5	6	7
5	7	9	9	9	8	3	6

Xóa phần tử cuối khỏi Dynamic Array



C++

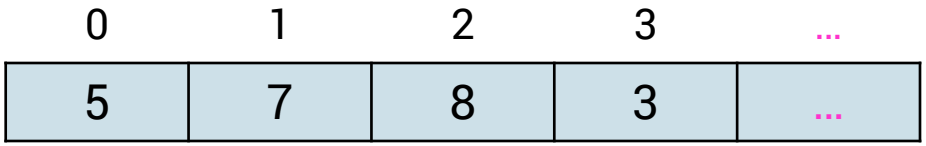
python

pop_back()

```
v.pop_back();
```

pop()

```
l.pop()
```



Xóa phần tử cuối của ArrayList

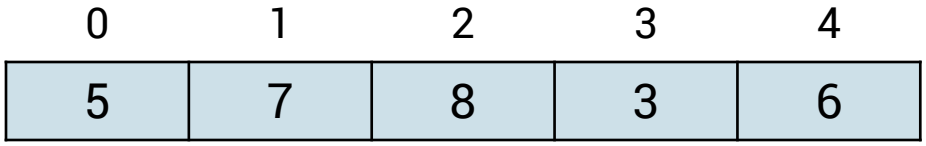
0	1	2	3	4
5	7	8	3	6

remove(index): Để xóa phần tử cuối cùng của ArrayList, ta sử dụng hàm remove để xóa. Lưu ý chương trình sẽ Throw exception nếu index không thỏa mãn giới hạn của ArrayList (index < 0 hoặc index >= size()).

```
a.remove(a.size()-1);
```

0	1	2	3	...
5	7	8	3	...

Xóa phần tử ở vị trí bất kỳ trong Dynamic Array



C++

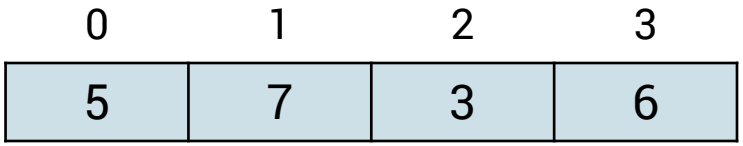
python

erase(iterator): Xóa **một** giá trị.

```
vector<int>::iterator it;
it = v.begin() + 2;
v.erase(it);
```

pop(pos): Xóa **giá trị** trong list ở vị trí bất kì và trả về giá trị bị xóa.

```
l.pop(2)
```



Xóa phần tử ở vị trí bất kì trong ArrayList

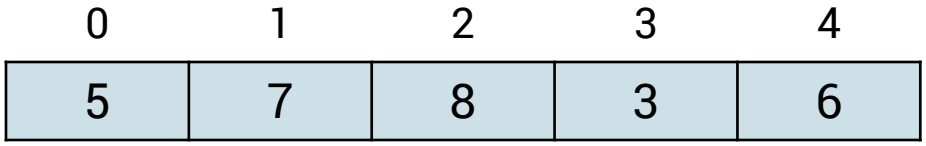
0	1	2	3	4
5	7	8	3	6

Như đã trình bày ở trên, để xóa phần tử bất trong Array List thì ta sử dụng lại hàm **remove(index).**

```
a.remove(2);
```

0	1	2	3
5	7	3	6

Xóa hàng loạt giá trị trong Dynamic Array



C++

erase(iterator1, iterator2)

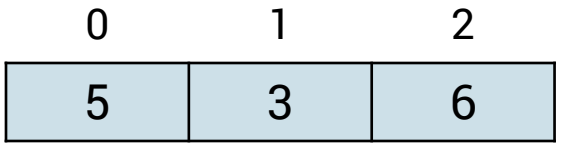
```
vector<int>::iterator it1;
vector<int>::iterator it2;
it1 = v.begin() + 1;
it2 = v.begin() + 3;
v.erase(it1, it2);
```

python

del <variable> [first: last]

Sẽ xóa các phần tử từ first đến trước last.

```
del l[1: 3]
```



Xóa hàng loạt giá trị trong Array List

0	1	2	3	4
5	7	8	3	6

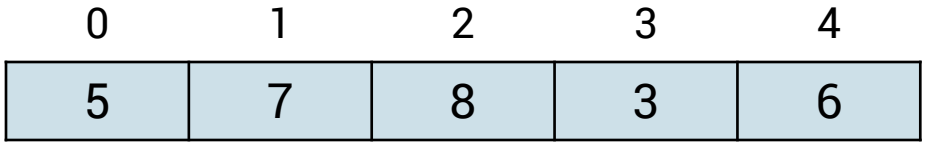
subList(fromIndex, toIndex).clear(): Kết hợp 2 hàm khác nhau để xóa 1 một đoạn trong ArrayList.

```
a.subList(1, 3).clear();
```

0	1	2
5	3	6

Lưu ý: *trong Java có thêm hàm removeRange(fromIndex, toIndex) cũng dùng để xóa, tuy nhiên đây là 1 hàm protected nên cần phải khai báo 1 lớp kế thừa và với 1 phương thức public removeRange để sử dụng.*

Xóa toàn bộ các phần tử trong Dynamic Array



C++

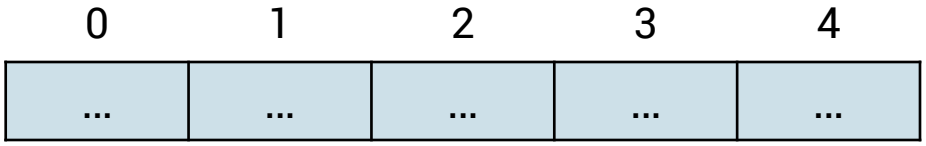
clear()

```
v.clear();
```

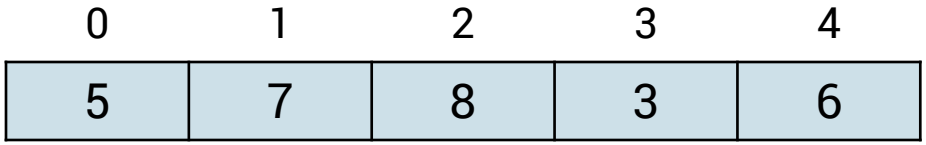
python™

clear()

```
l.clear()
```

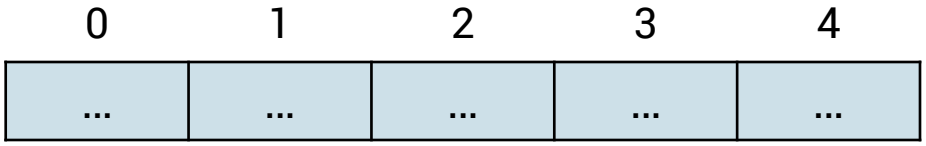


Xóa toàn bộ ArrayList

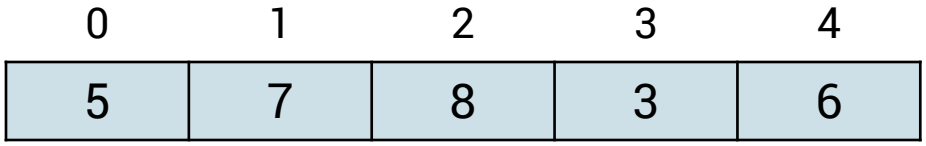


clear(): Xóa tất cả phần tử trong ArrayList.

```
a.clear();
```



Lấy kích thước của Dynamic Array



C++

size()

```
int n = v.size();
cout<<n;
```

python™

len(obj)

```
n = len(l)
print(n)
```

5

Lấy kích thước của ArrayList

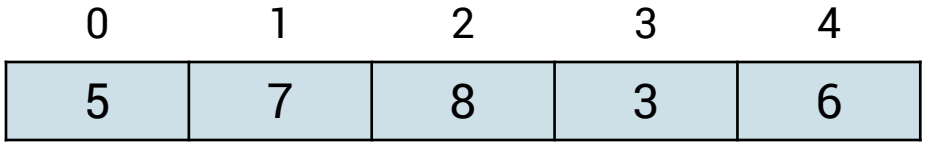
0	1	2	3	4
5	7	8	3	6

size(): Lấy số lượng phần tử trong ArrayList.

```
a.size();
```

5

Thay đổi kích thước Dynamic Array lớn thêm

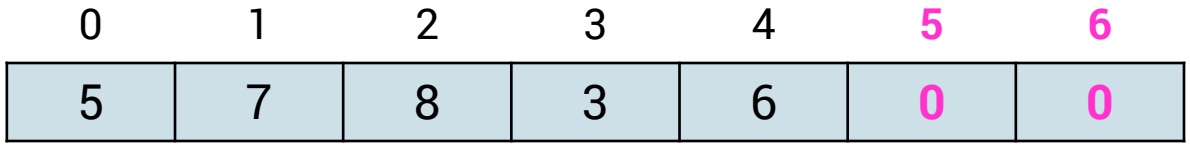


resize(size_type): Thay đổi và gán giá trị bằng giá trị mặc định của kiểu dữ liệu **nếu có thể**.

```
v.resize(7);
```

extend(list): nối 2 list lại với nhau.

```
l.extend(2*[0])
```



Thay đổi kích thước ArrayList lớn thêm

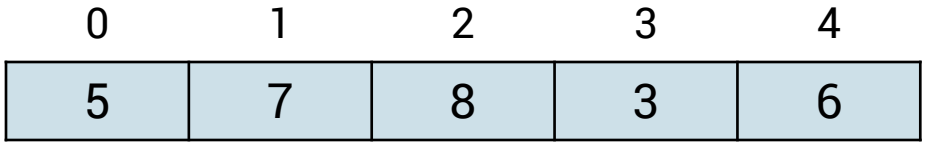
0	1	2	3	4
5	7	8	3	6

Java không có hàm resize mà phải tự cài đặt bằng tay.

```
for (int i = 5; i < 7; i++) {
    a.add(0);
}
```

0	1	2	3	4	5	6
5	7	8	3	6	0	0

Thay đổi kích thước Dynamic Array nhỏ lại

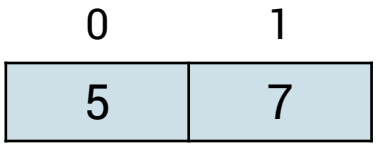


C++

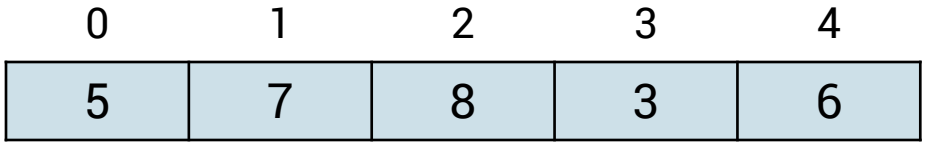
```
v.resize(2);
```

python

```
l = l[0:2]
```

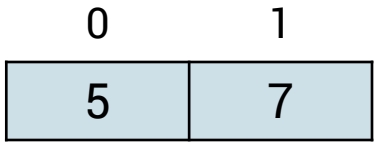


Thay đổi kích thước ArrayList nhỏ lại

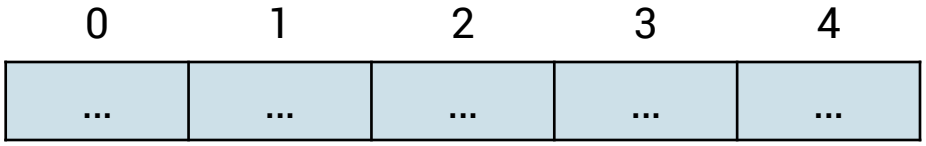


Java không có hàm resize mà phải tự cài đặt bằng tay.

```
a.subList(2, 5).clear();
```



Kiểm tra xem Dynamic Array có rỗng không



empty()

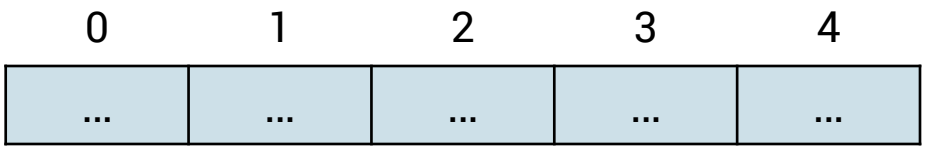
```cpp
vector<int> v;
if(v.empty() == true)
    cout<<"DA is empty!";
else
    cout<<"DA is not empty!";
```

Sử dụng lại hàm len

```python
l = []
if len(l) == 0:
    print("DA is empty")
else:
    print("DA is not empty")
```

DA is empty!

Kiểm tra ArrayList có rỗng không



isEmpty(): Kiểm tra ArrayList có rỗng hay không.

```
ArrayList<Integer> a = new ArrayList<Integer>();
if (a.isEmpty() == true)
    System.out.println("ArrayList is empty!");
else
    System.out.println("ArrayList is not empty!");
```

ArrayList is empty!

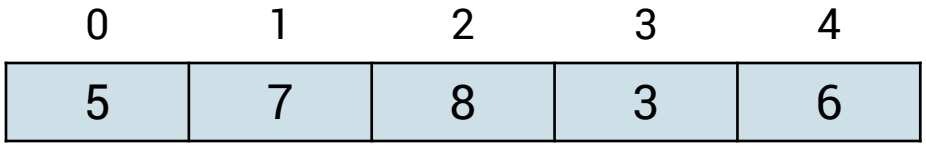
Một số hàm thành viên khác:

contains(value): Trả về true nếu giá trị đó có trong ArrayList, ngược lại trả về false.

indexOf(value): Trả về vị trí phần tử nếu có phần tử trong ArrayList bằng với giá trị “value” trong ArrayList, ngược lại trả về -1.

CÁC LƯU Ý KHI SỬ DỤNG DYNAMIC ARRAY

Truy cập ngẫu nhiên vào Dynamic Array



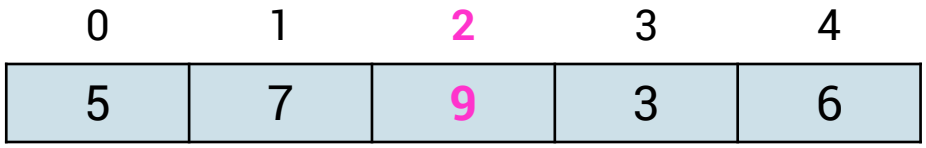
Có thể truy cập vào thành phần Dynamic Array để thay đổi hoặc lấy giá trị nếu chỉ số hợp lệ.

C++

```
v[2] = 9;
int a = v[2];
cout<<a;
```

python

```
l[2] = 9
a = l[2]
print(a)
```



9

Truy cập ngẫu nhiên vào ArrayList

0	1	2	3	4
5	7	8	3	6

Trong Java không có toán tử [ ] giống C++ và Python vì thế dùng 2 hàm set(index, value) và get(index) để cập nhật và lấy giá trị của một phần tử trong ArrayList.

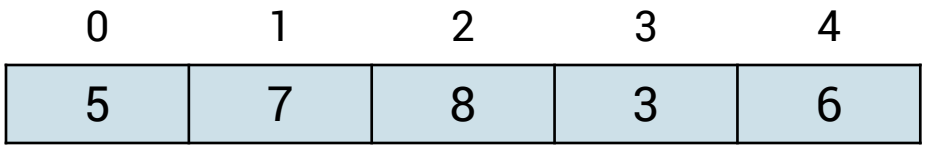
Throw exception nếu index không thỏa mãn giới hạn của ArrayList (index < 0 || index > size()).

```
a.set(2, 9);
Integer value = a.get(2);
System.out.print(value);
```

0	1	2	3	4
5	7	9	3	6

9

Duyệt xuôi trong Dynamic Array



C++

```
for(int i=0; i<v.size(); i++)
{
    cout<<v[i]<<", ";
}
```

```
vector<int>::iterator it;
for(it=v.begin(); it!=v.end(); it++)
{
    cout<<*it<<", ";
}
```

python

```
for i in range(0, len(l)):
    print(l[i],end=', ')
```

5, 7, 8, 3, 6

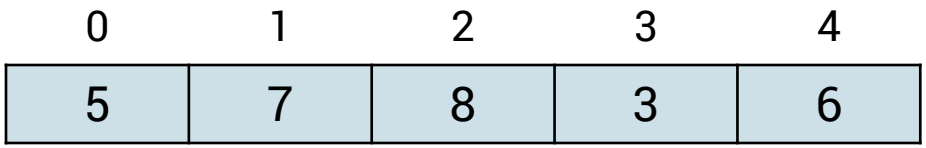
Duyệt xuôi trong ArrayList

0	1	2	3	4
5	7	8	3	6


```
for (int i = 0; i < a.size(); i++) {
    System.out.print(a.get(i) + ", ");
}
```

5, 7, 8, 3, 6

Duyệt ngược trong Dynamic Array



C++

```
for(int i=v.size(); i>=0; i--){
    cout<<v[i]<<", ";
}
```

```
ector<int>::reverse_iterator it;
for(it=v.rbegin(); it!=v.rend(); it++) {
    cout<<*it<<", ";
}
```

python

```
for i in range(len(l)-1, -1, -1):
    print(l[i],end=', ')
```

6, 3, 8, 7, 5

Duyệt ngược trong ArrayList

0	1	2	3	4
5	7	8	3	6


```
for (int i = a.size() - 1; i >= 0; i--) {
    System.out.print(a.get(i) + ", ");
}
```

6, 3, 8, 7, 5

Sử dụng vector như mảng 2 chiều

Khai báo:

```
vector<vector<data_type> > variable_name;
```

Ví dụ:

```
vector<vector<int> > v;
```

```
void Input(vector<vector<int> > &v, int &m, int &n) {
    cin>>m>>n;
    v.resize(m);
    for(int i=0; i<m; i++) {
        v[i].resize(n);
        for(int j=0; j<n; j++) {
            cout<<"v["<<i<<"]"<<"["<<j<<"]: ";
            cin>>v[i][j];
        }
    }
}
```

Sử dụng list như mảng 2 chiều

List là kiểu dữ liệu động, mỗi phần tử có thể là 1 list khác, 1 giá trị,... Vì vậy có thể sử dụng như mảng 2 chiều. Khai báo vẫn như bình thường, tuy nhiên ở mỗi phần tử thì sẽ khai báo nó thành 1 list (list lồng list).

```
def inputArray2D(arr2d):
    n, m = map(int, input().split())
    for i in range(n):
        arr2d.append([])
        arr2d[i] = list(map(int, input().split()))
    return n, m
```

Sử dụng ArrayList như mảng 2 chiều

ArrayList là kiểu dữ liệu động, mỗi phần tử có thể là 1 ArrayList khác, 1 giá trị, … Vì vậy có thể sử dụng như mảng 2 chiều. Khai báo vẫn như bình thường, tuy nhiên ở mỗi phần tử thì sẽ khai báo nó thành 1 ArrayList (list lồng list).

```
ArrayList<ArrayList<Integer>> arrList2d
= new ArrayList<ArrayLIst<Integer>> ();
```

Hỏi đáp