



VECTORS

v

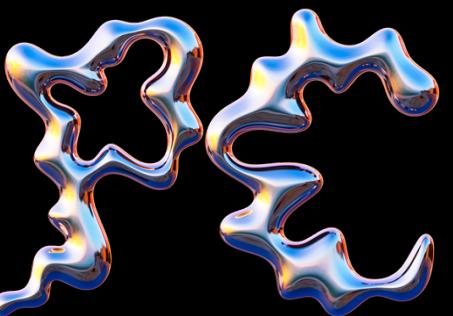
Element 1

Element 2

Element 3

...

Element N



CΦ

Vectors

- Vectors are used to store elements of similar data types. We can change the size of the vector during the execution of a program as per our requirements.
- The ability to resize itself, when an element is inserted or deleted with their storage being handled automatically.

Why vectors?

Vectors are commonly used to take inputs, store and process collections of data, such as a list of numbers, strings, objects, or any other type of element. They provide dynamic resizing capabilities, making them suitable for scenarios where the size of the collection may change during runtime.

Declaration and Initialization:

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    vector<int> v;
}
```

1) `vector<int> v(n)`

2) `vector<int>v(size,value)`

OR

```
vector <int> v;  
v.assign(size,value)
```

3) `vector<int> v{ 10, 20, 30 };`

size - number of values to be assigned
value - value to be assigned to the vector

Vectors can be of any data type:

```
vector<data_type> variable;
```

```
vector<float> v1(10);
```

```
vector<char> v2(5);
```

```
vector<bool> v3(7);
```

Index and Position

An index refers to an element's position within an ordered list, like a vector. The first element has an index of 0
The index of a number is always 1 less than it's position.

Indexes	0	1	2	3	4
Values	1	3	8	23	99

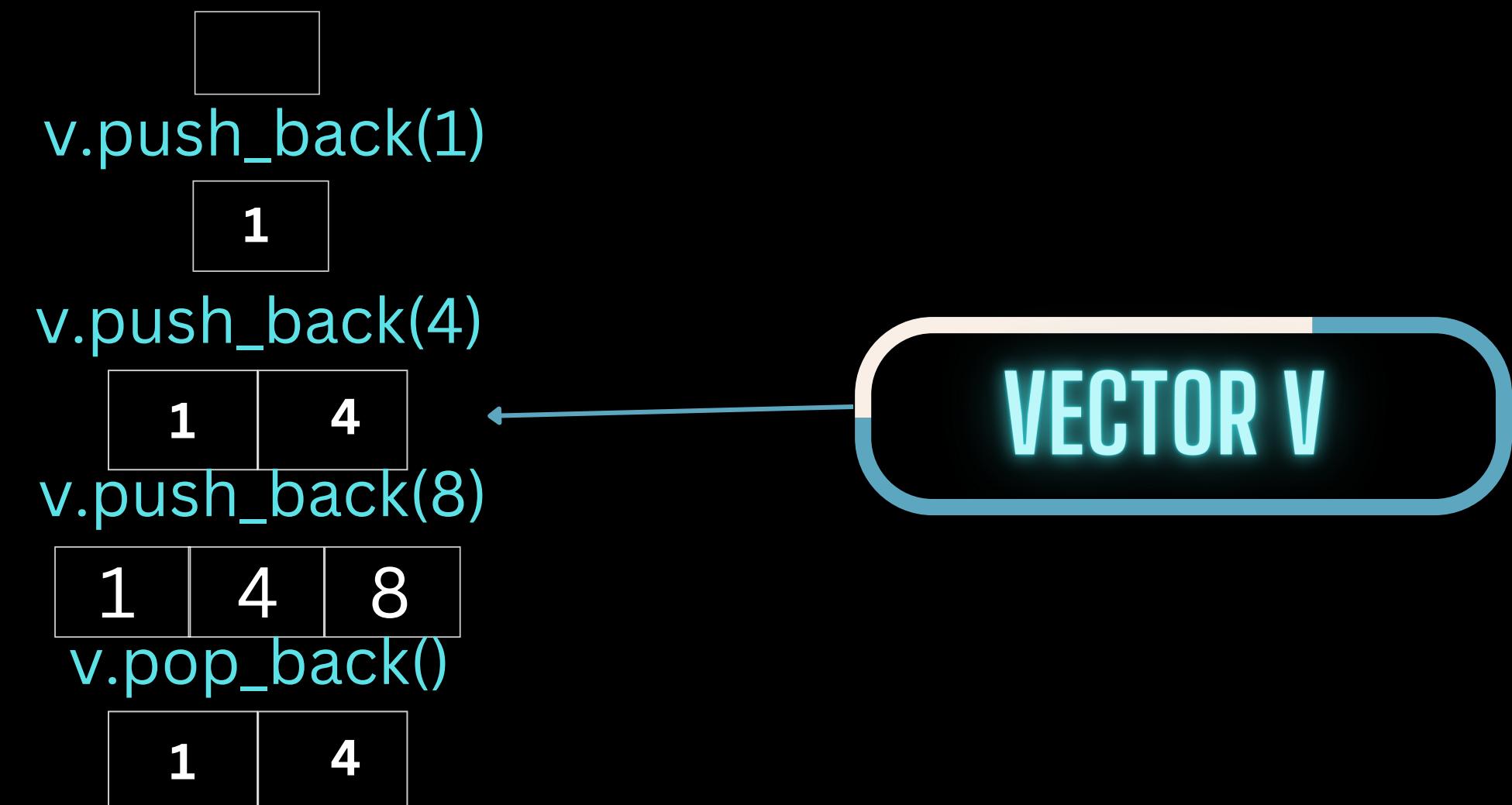
1st element (position)

2nd element (position)

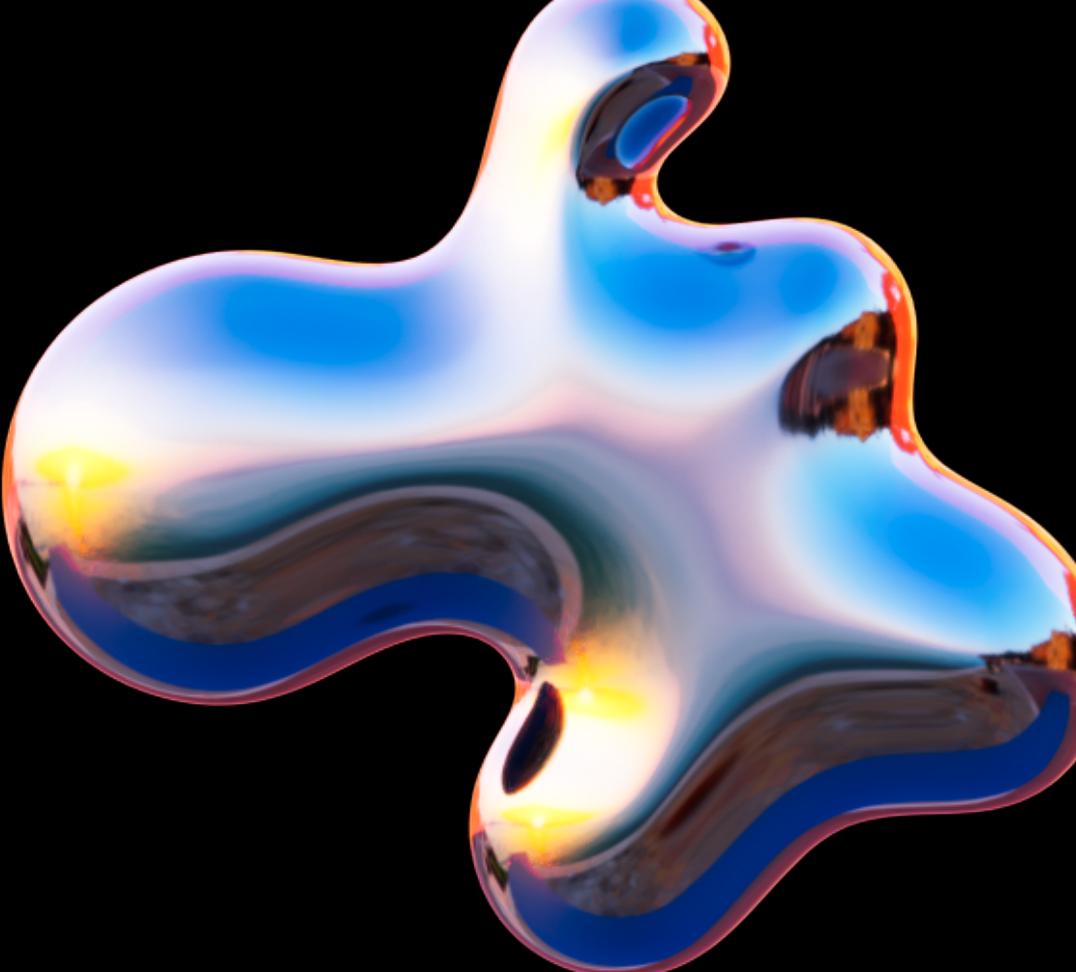
COMMON FUNCTIONS

- 1. **V.push_back(a):**
 - USE: Inserts an element a at the end of the vector v.
- 2. **V.pop_back():**
 - USE: Removes last element of the vector v.
- 3. **V.size():**
 - USE: Returns the number of elements in the vector v.
- 4. **v[i]**
 - USE: Returns the element at (i)th index in vector v.

```
vector<int>v;  
v.push_back(1);  
v.push_back(4);  
v.push_back(8);  
v.pop_back();
```



```
vector<int>v;
//v.size() returns the number of elements in v
cout<<"initial size: "<<v.size()<<endl;
// v.push_back() adds an element to the end of v
v.push_back(3);
v.push_back(4);
cout<<"size after push_back: "<<v.size()<<endl;
cout<<"elements of v: ";
for(int i=0;i<v.size();i++)
cout<<v[i]<<" ";
cout<<endl;
// v.pop_back() removes the last element
v.pop_back();
cout<<"size after pop_back: "<<v.size()<<endl;
cout<<"elements of v: ";
for(int i=0;i<v.size();i++)
cout<<v[i]<<endl;
```



initial size: 0
size after push_back: 2
elements of v: 3 4
size after pop_back: 1
elements of v: 3



5. v.clear():

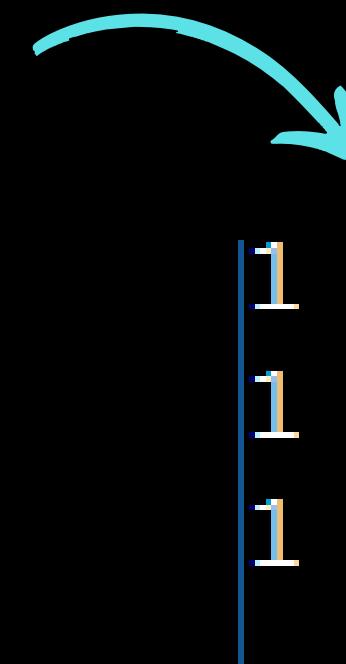
- USE: Erases the vector v.

6) v.resize(n)

v.resize(n,val)

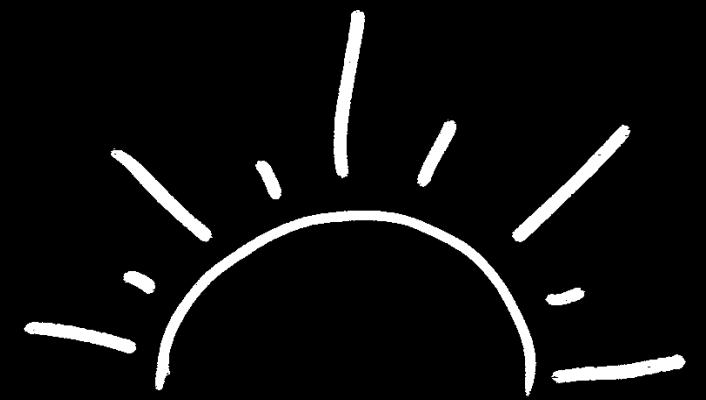
- If the given value of n is less than the current size of vector then extra elements are removed.
- If n is more than current size of vector then upcoming elements are appended at the end of the vector.

```
vector<int> v {1,2,3,4,5};  
v.resize(4);  
for (int i = 0; i < v.size(); i++)  
cout << v[i] << " ";  
cout<<endl;  
v.resize(6);  
for (int i = 0; i < v.size(); i++)  
cout << v[i] << " ";  
cout<<endl;  
v.resize(8, 9);  
for (int i = 0; i < v.size(); i++)  
cout << v[i] << " ";
```



Let's Implement

The best way to learn



The Warm Up

Problem

You are given a string of length n consisting only of the characters '+' and '-'. If the character is '-', you add one more '-' to its right and if it's '+', you add '+' to the right as well as left. You are asked to print the final reversed string.

`reverse(v.begin(), v.end())`

THANK YOU!