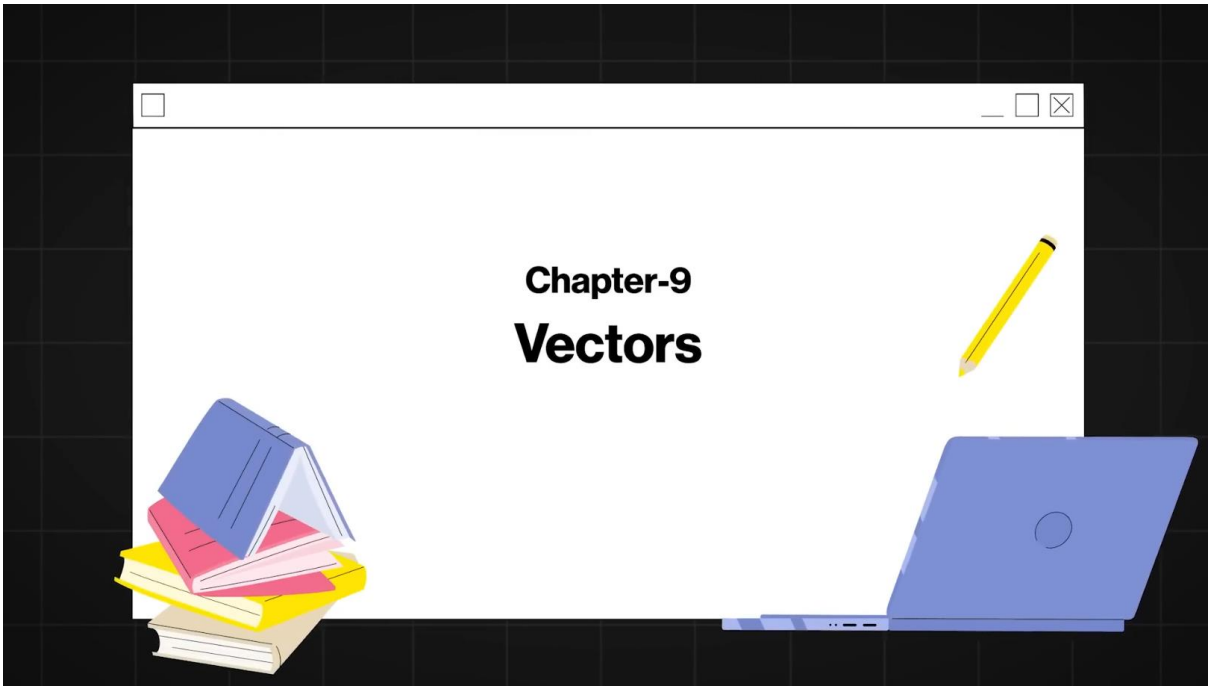
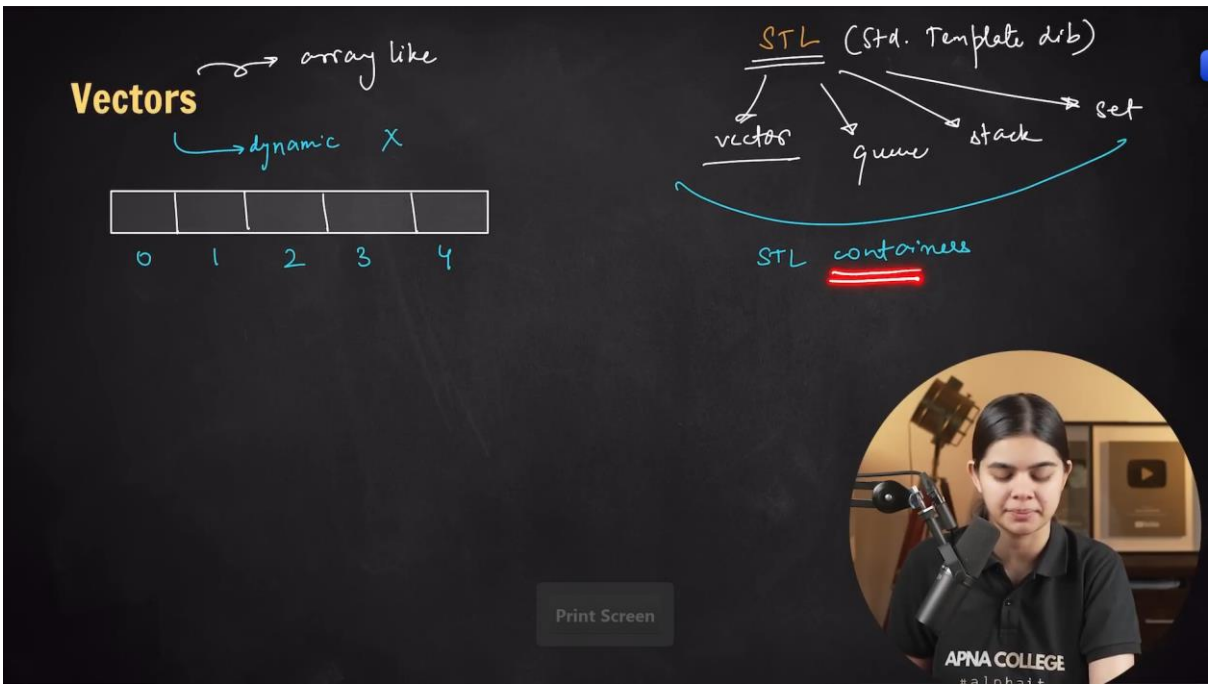
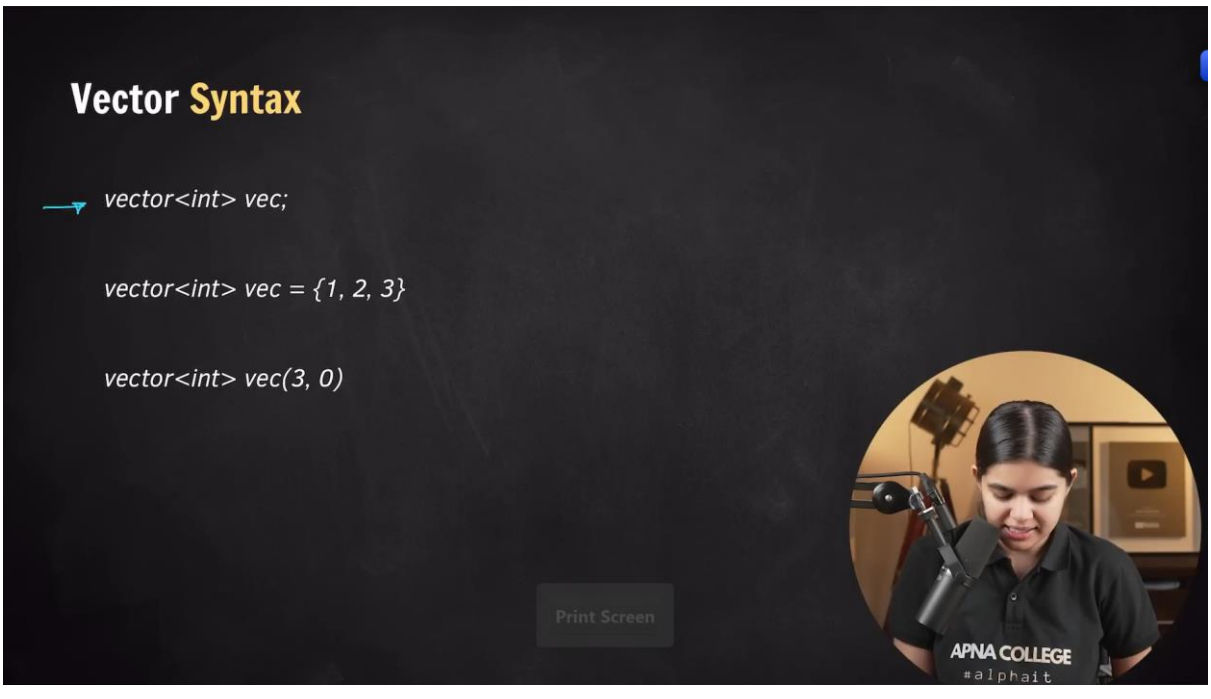
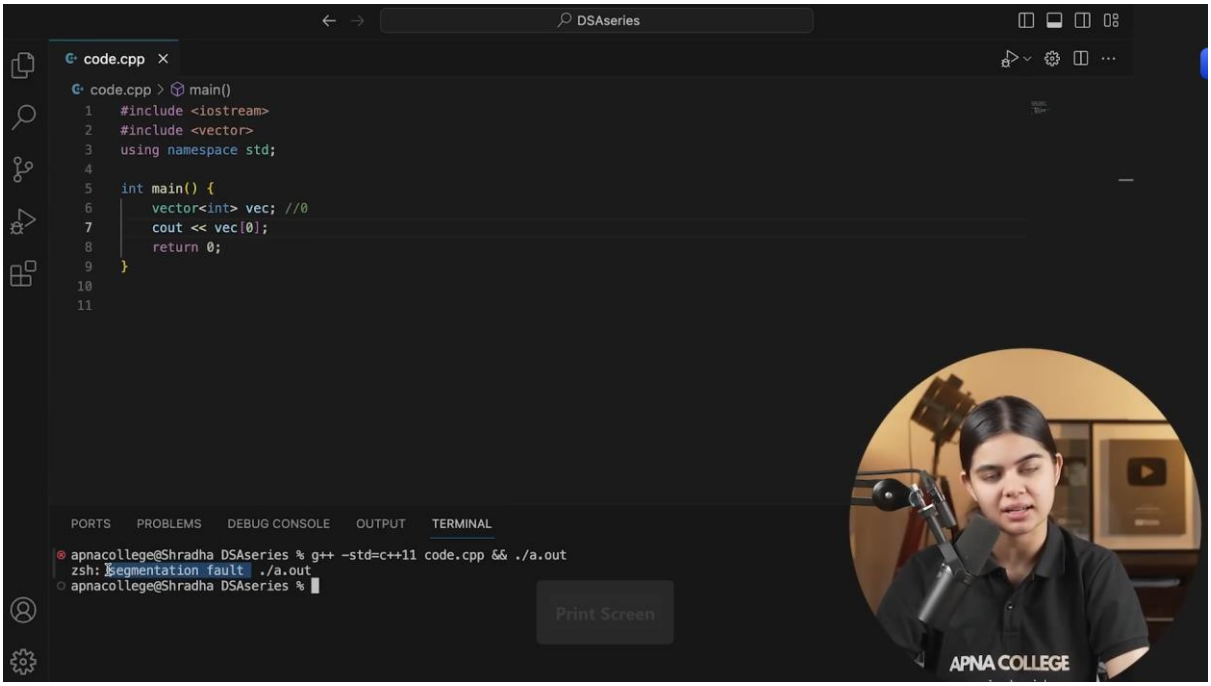
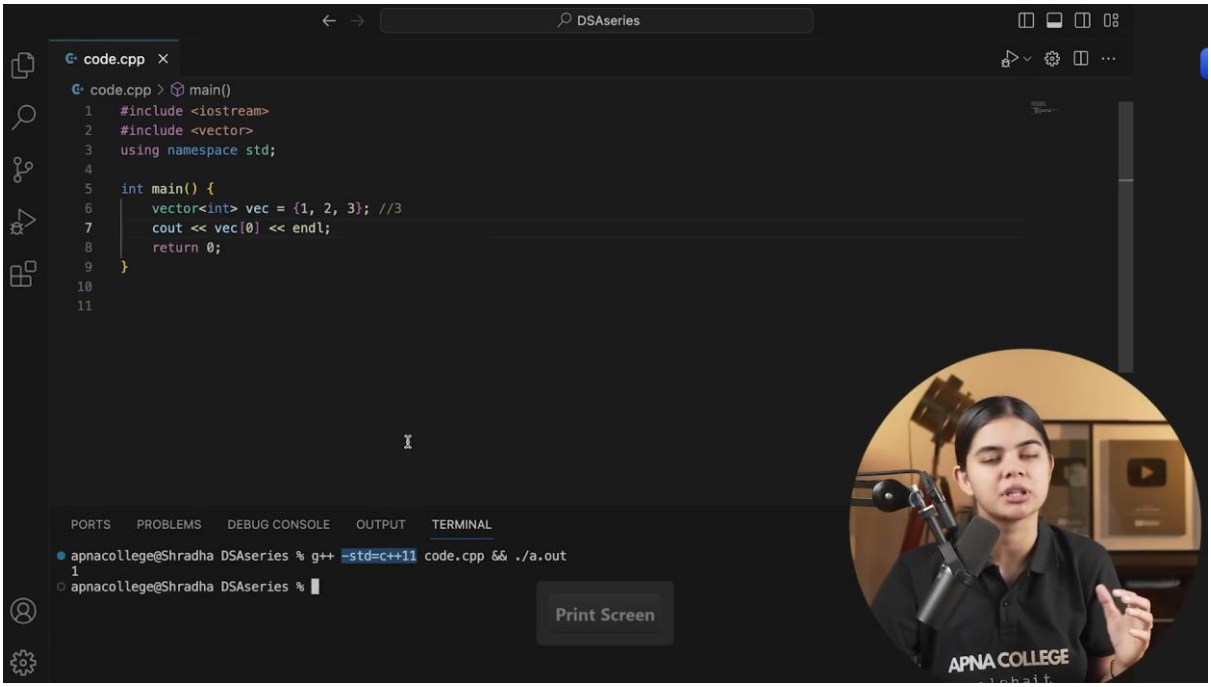
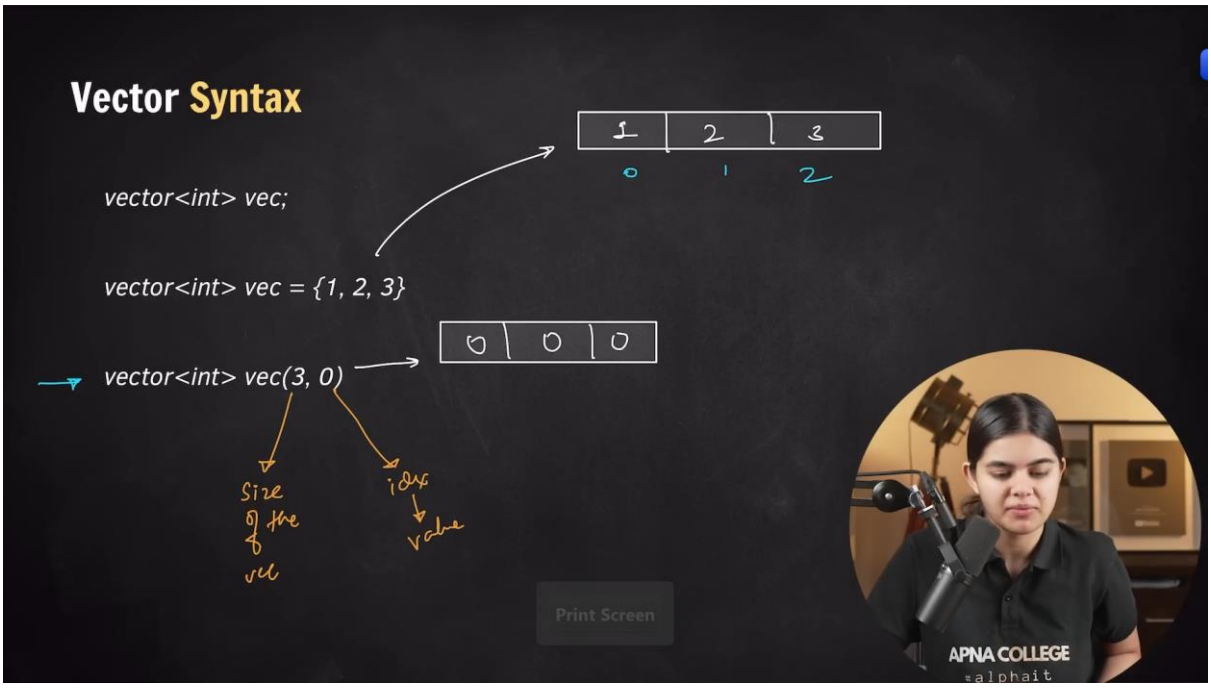

Chapter-9
Vectors


Vectors
array like
dynamic X
0 1 2 3 4
STL (Std. Template lib)
vector
queue
stack
set
STL containers
Print Screen
APNA COLLEGE


Vector Syntax
vector<int> vec;
vector<int> vec = {1, 2, 3}
vector<int> vec(3, 0)
Print Screen
APNA COLLEGE
#alphait


DSAseries
code.cpp
code.cpp > main()
#include <iostream>
#include <vector>
using namespace std;
int main() {
vector<int> vec; //0
cout << vec[0];
return 0;
}
PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL
apnacollege@Shradha DSAseries % g++ -std=c++11 code.cpp && ./a.out
zsh: segmentation fault ./a.out
apnacollege@Shradha DSAseries %
Print Screen
APNA COLLEGE


DSAseries
code.cpp
code.cpp > main()
#include <iostream>
#include <vector>
using namespace std;
int main() {
vector<int> vec = {1, 2, 3}; //3
cout << vec[0] << endl;
return 0;
}
PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL
apnacollege@Shradha DSAseries % g++ -std=c++11 code.cpp && ./a.out
1
apnacollege@Shradha DSAseries %
Print Screen
APNA COLLEGE


Vector Syntax
vector<int> vec;
vector<int> vec = {1, 2, 3}
vector<int> vec(3, 0)
1 2 3
0 1 2
0 0 0
Size of the vec
idx
value
Print Screen
APNA COLLEGE
alphait

DSAseries

code.cpp

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<char> vec = {'a', 'b', 'c', 'd', 'e'};

    for(char val : vec) {
        cout << val << endl;
    }

    return 0;
}
```

PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL

```
apnacollege@Shradha DSAseries % g++ -std=c++11 code.cpp && ./a.out
a
b
c
d
e
apnacollege@Shradha DSAseries %
```

Print Screen

Vector Syntax

for each loop

for (int i : vec)

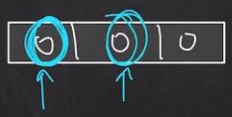
idx value

idx

vector<int> vec;

vector<int> vec = {1, 2, 3}

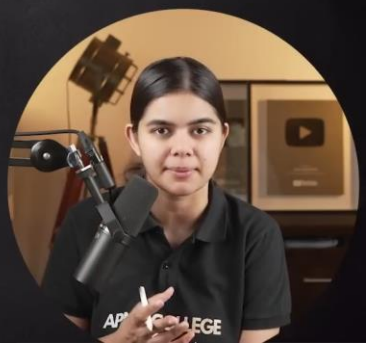
vector<int> vec(3, 0)



Size of the vec

idx value

Print Screen



DSAseries

code.cpp

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<int> vec;

    cout << "size = " << vec.size() << endl;

    for(char val : vec) { //for each loop
        cout << val << endl;
    }

    return 0;
}
```

PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL

```
apnacollege@Shradha DSAseries % g++ -std=c++11 code.cpp && ./a.out
size = 0
apnacollege@Shradha DSAseries %
```

Print Screen

DSAseries

code.cpp

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<int> vec;

    cout << "size = " << vec.size() << endl;
    vec.push_back(25);
    vec.push_back(35);
    vec.push_back(45);
    cout << "after push back size = " << vec.size() << endl;

    for(int val : vec) { //for each loop
        cout << val << endl;
    }

    return 0;
}
```

PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL

```
apnacollege@Shradha DSAseries % g++ -std=c++11 code.cpp && ./a.out
size = 0
after push back size = 3
25
35
45
apnacollege@Shradha DSAseries %
```

Print Screen

DSAseries

code.cpp

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<int> vec;

    vec.push_back(25);
    vec.push_back(35);
    vec.push_back(45);

    cout << "after push back size = " << vec.size() << endl;

    vec.pop_back(); //45

    cout << vec.back() << endl;

    return 0;
}
```

PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL

```
apnacollege@Shradha DSAseries % g++ -std=c++11 code.cpp && ./a.out
after push back size = 3
25
apnacollege@Shradha DSAseries % g++ -std=c++11 code.cpp && ./a.out
after push back size = 3
35
apnacollege@Shradha DSAseries %
```

Print Screen

DSAseries

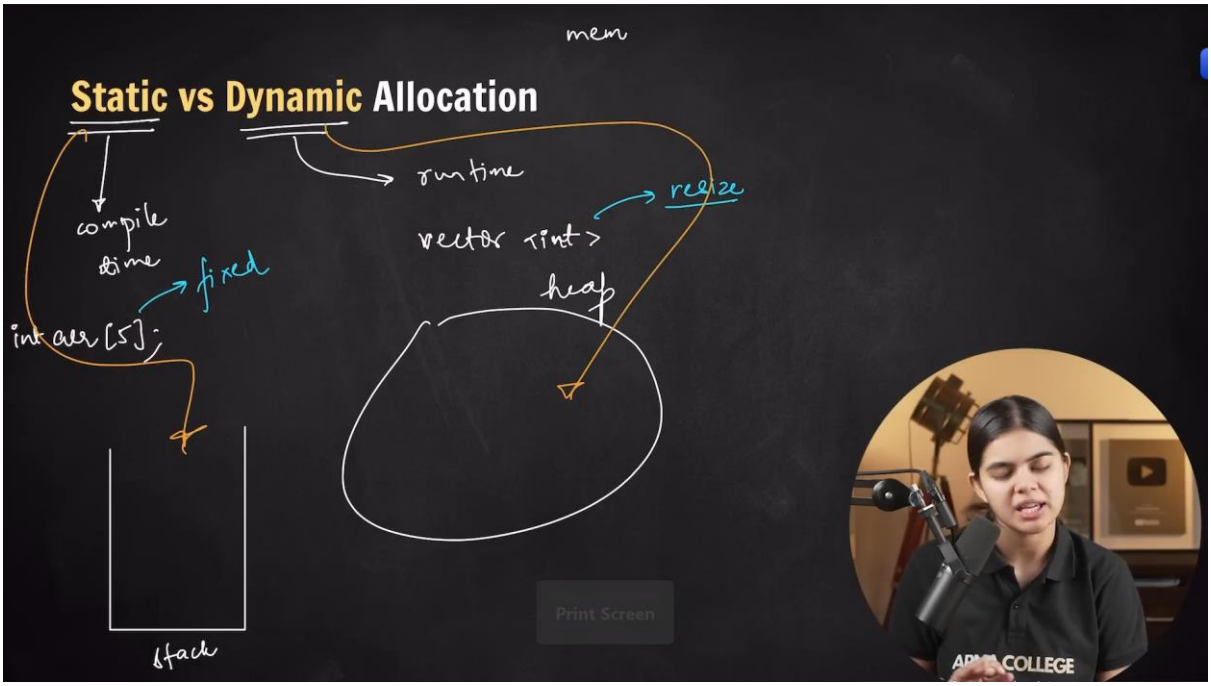
code.cpp

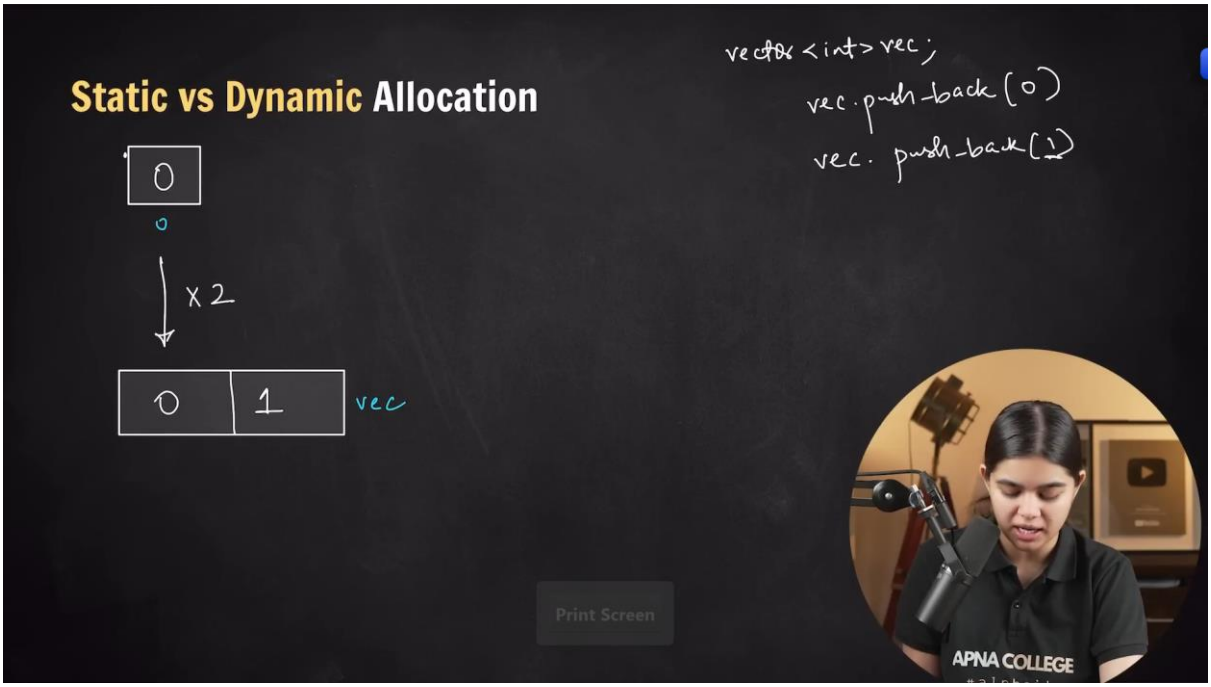
```cpp
#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<int> vec;

    vec.push_back(25);
    vec.push_back(35);
    vec.push_back(45);

    cout << "after push back size = " << vec.size() << endl;

    vec.pop_back(); //45

    cout << vec.at(1) << endl;

    return 0;
}
```

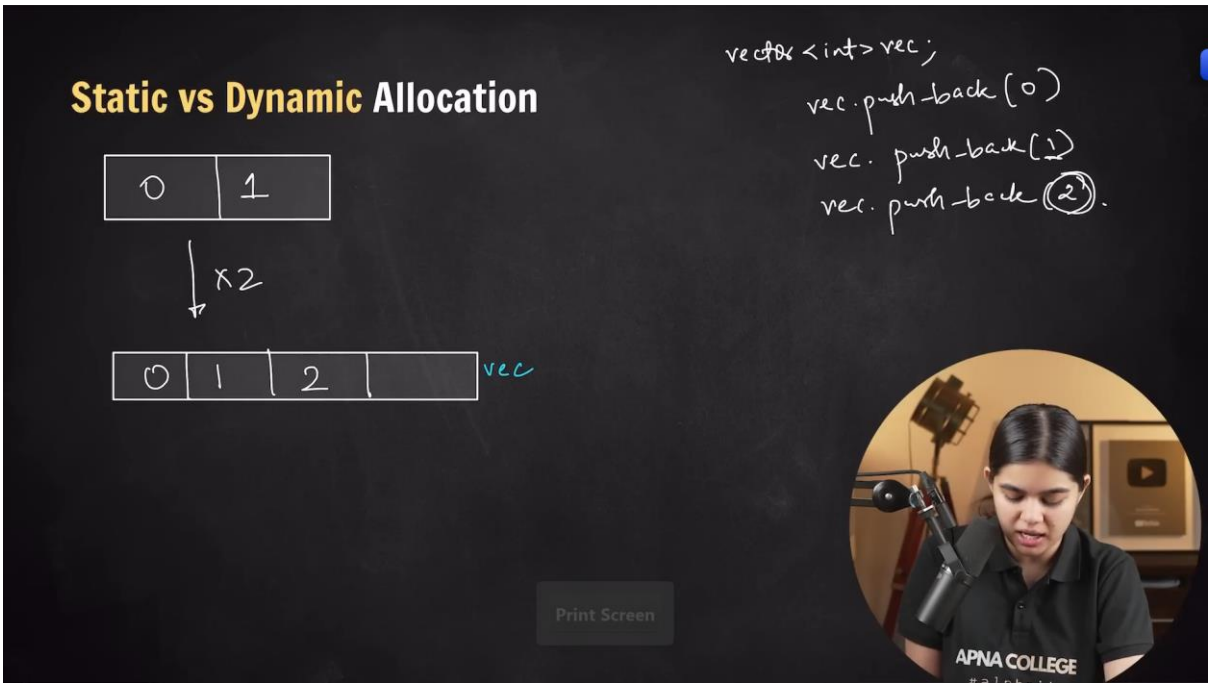
PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL

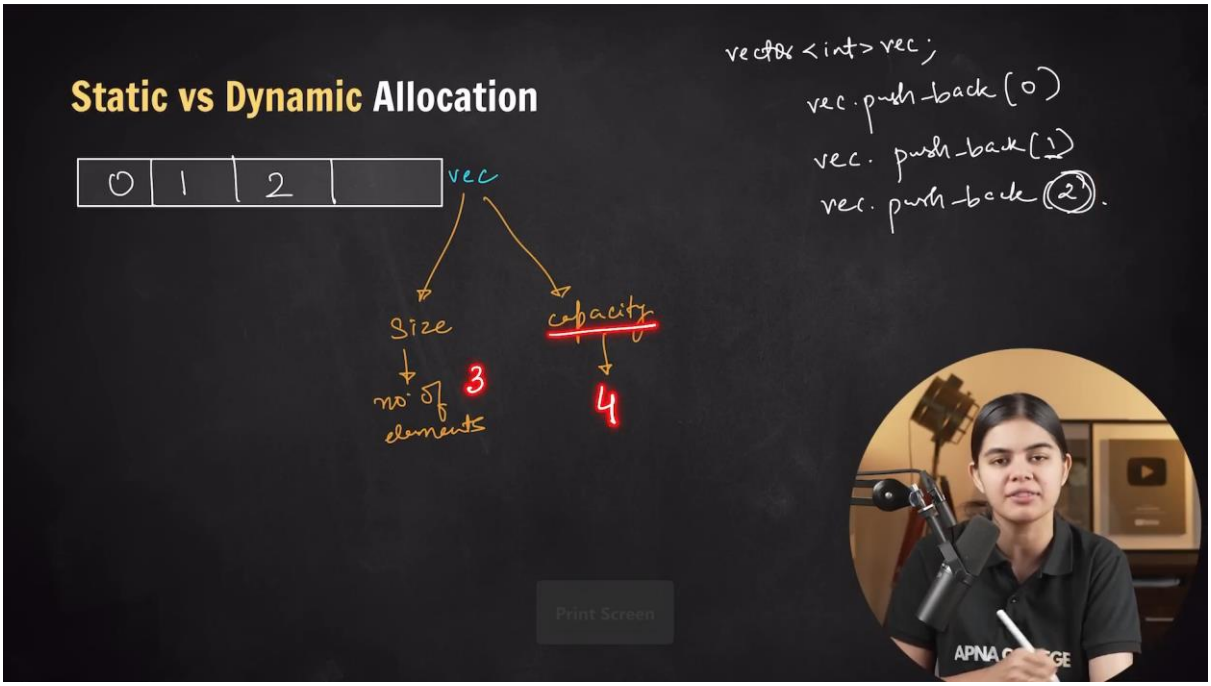
```
apnacollege@Shradha DSAseries % g++ -std=c++11 code.cpp && ./a.out
after push back size = 3
25
apnacollege@Shradha DSAseries % g++ -std=c++11 code.cpp && ./a.out
after push back size = 3
35
apnacollege@Shradha DSAseries %
```

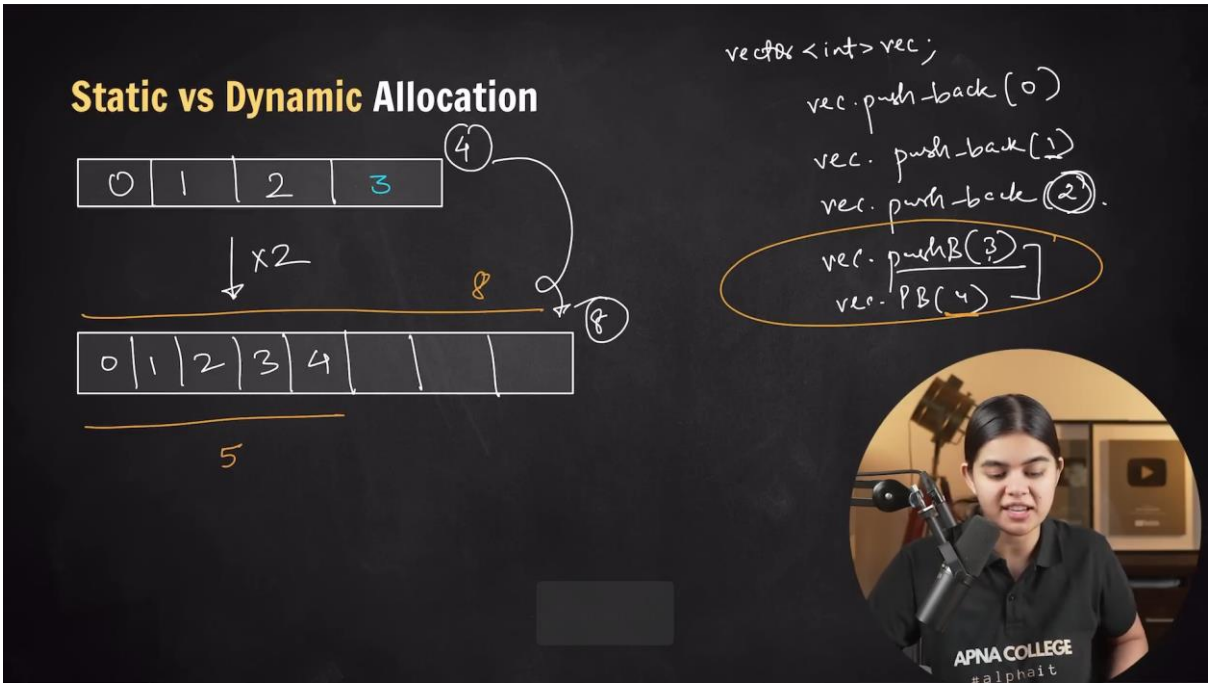
Print Screen

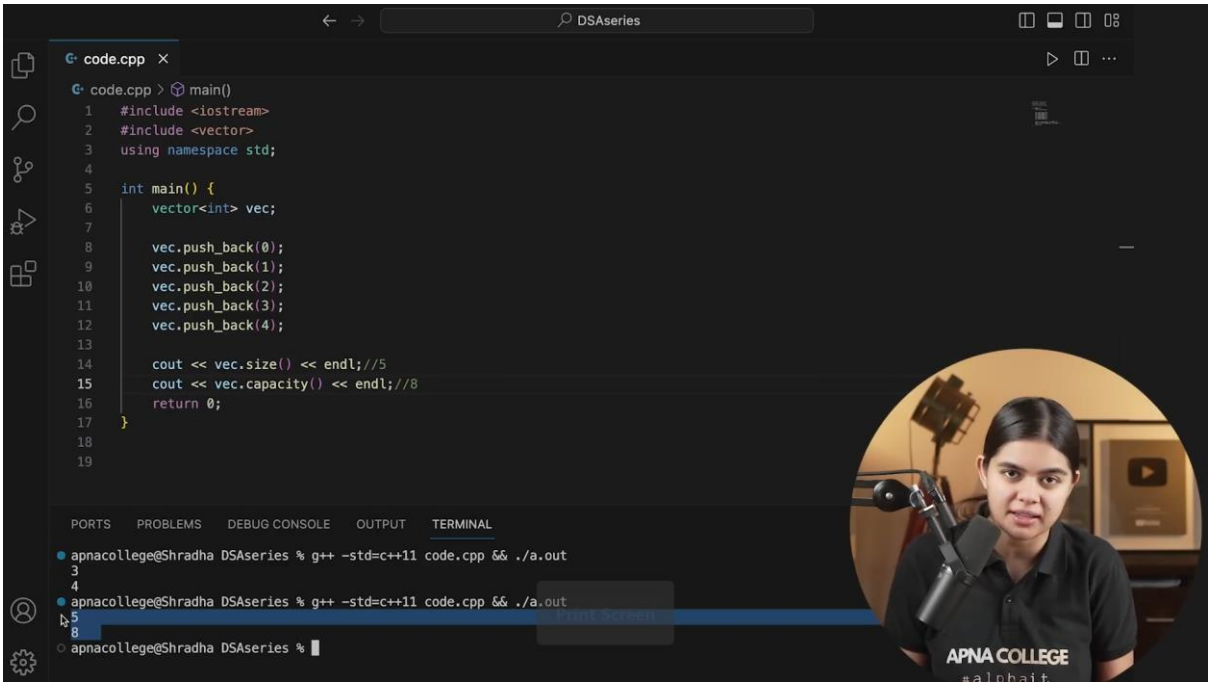

mem
Static vs Dynamic Allocation
compile time
runtime
fixed
resize
vector <int>
int arr[5];
heap
stack
Print Screen
APNA COLLEGE

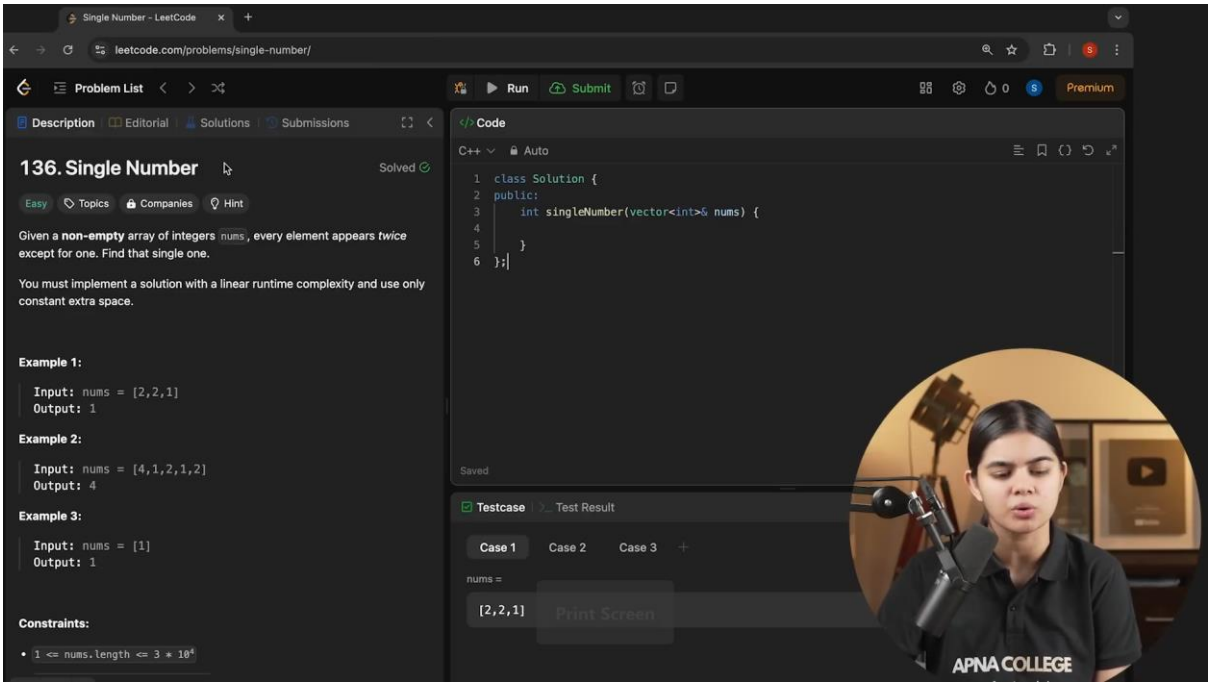

Static vs Dynamic Allocation
vector <int> vec;
vec.push_back(0)
vec.push_back(1)
0
x2
0
1
vec
Print Screen
APNA COLLEGE

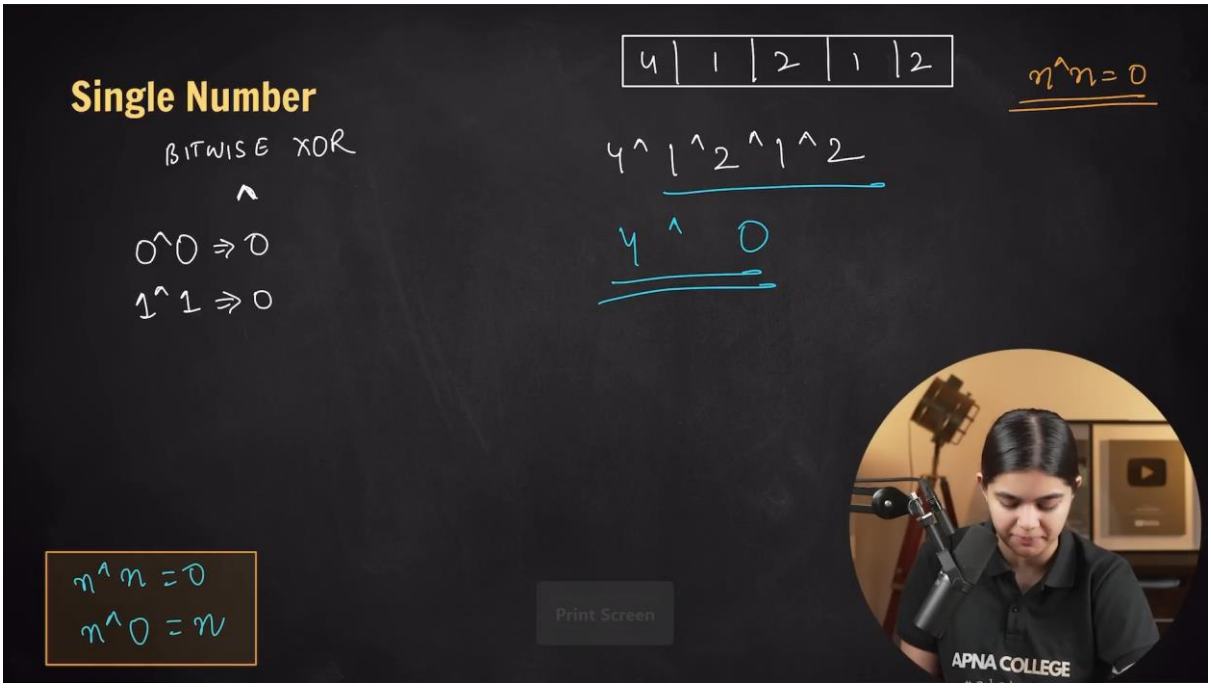

Static vs Dynamic Allocation
vector <int> vec;
vec.push_back(0)
vec.push_back(1)
vec.push_back(2).
0
1
x2
0
1
2
vec
Print Screen
APNA COLLEGE

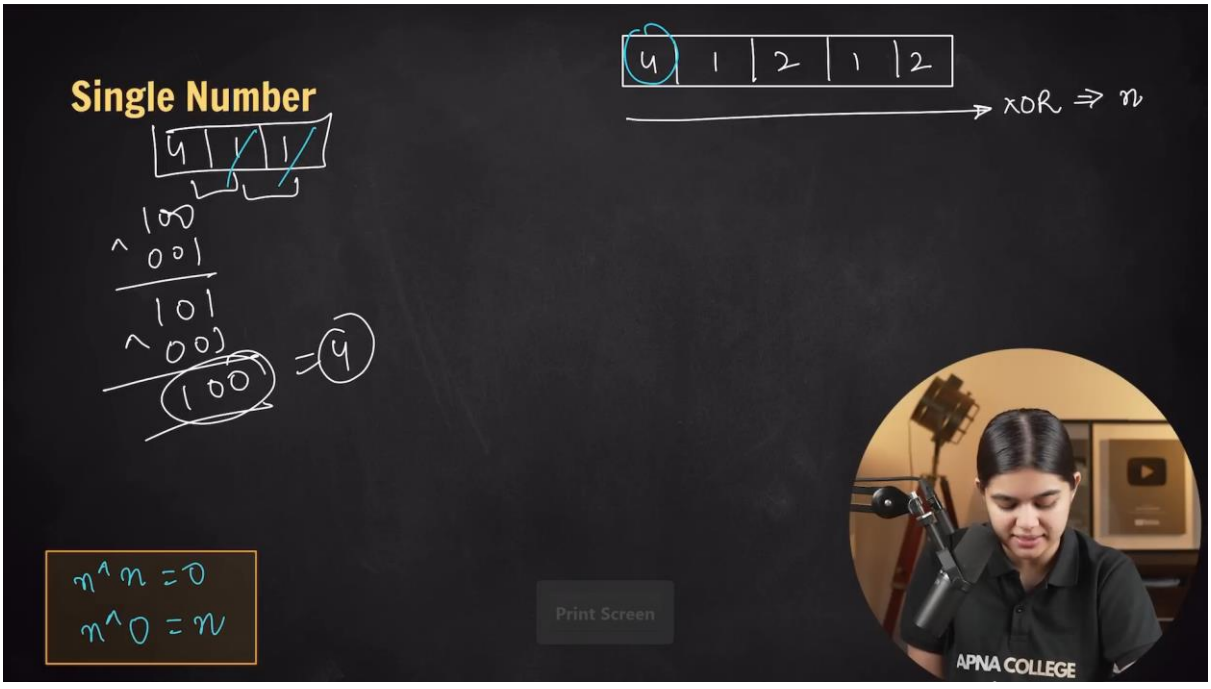

Static vs Dynamic Allocation
vector<int> vec;
vec.push_back(0)
vec.push_back(1)
vec.push_back(2).
0 1 2
vec
Size
no. of elements
3
capacity
4
APNA COLLEGE

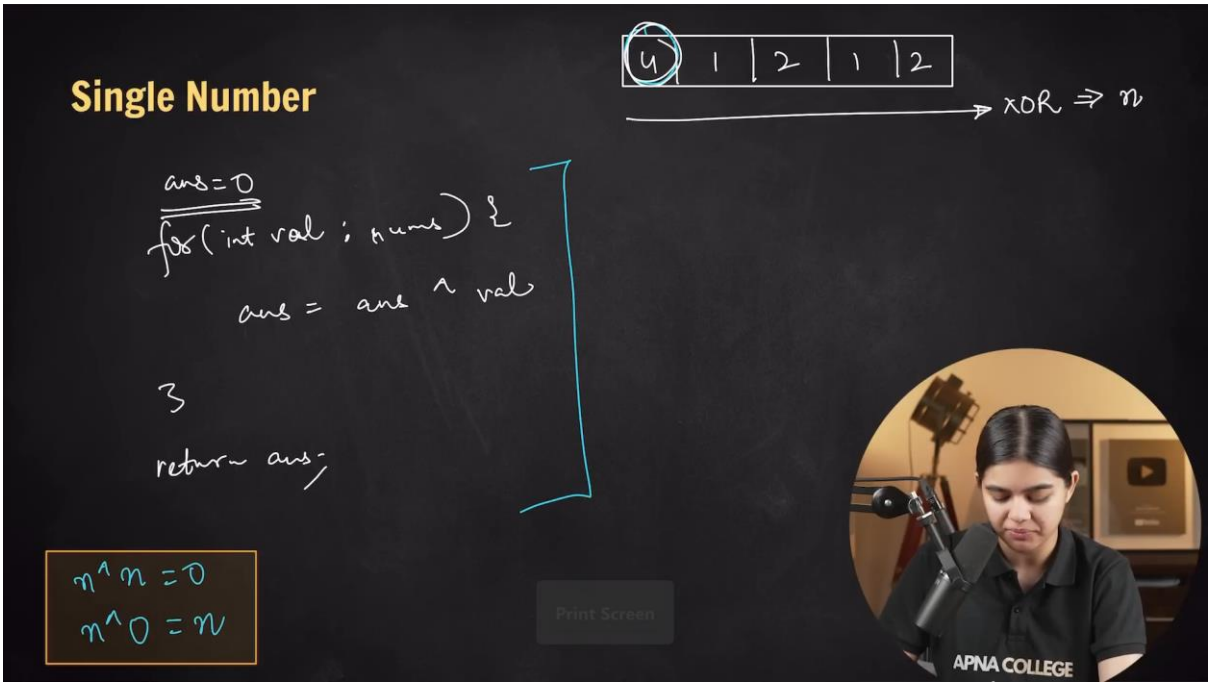

Static vs Dynamic Allocation
vector<int> vec;
vec.push_back(0)
vec.push_back(1)
vec.push_back(2).
vec.pushB(3)
vec.PB(4)
0 1 2 3
4
x2
8
8
0 1 2 3 4
5
APNA COLLEGE

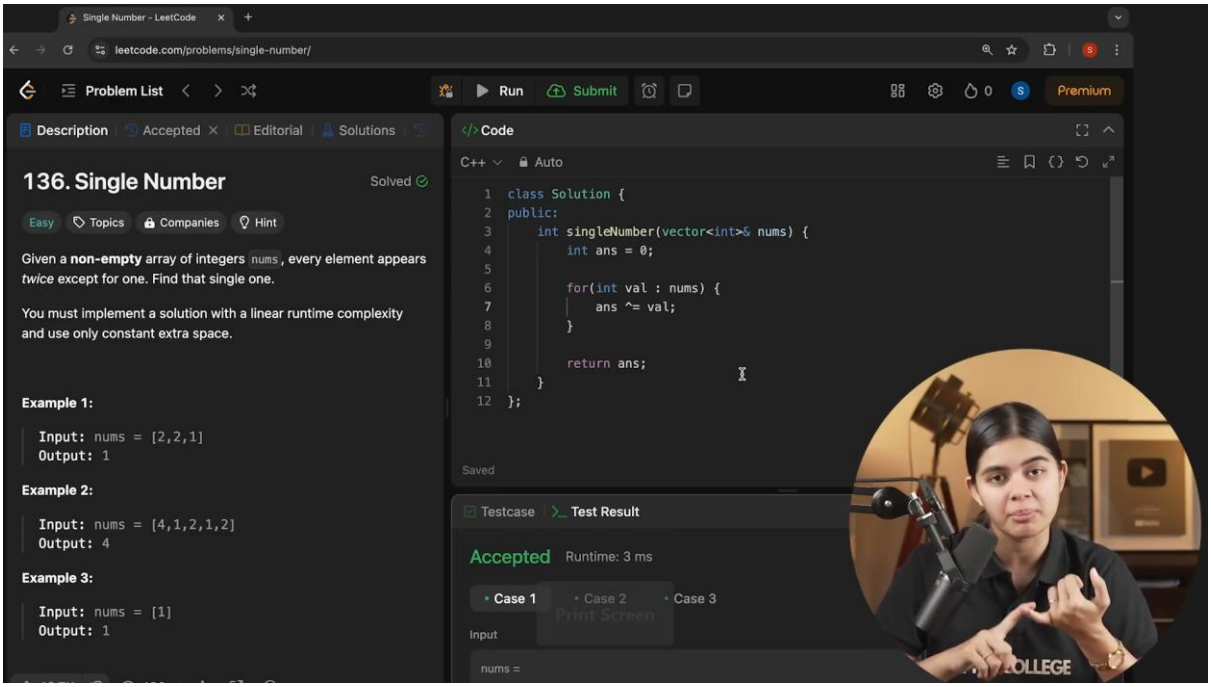

DSAseries
code.cpp
code.cpp > main()
#include <iostream>
#include <vector>
using namespace std;
int main() {
vector<int> vec;
vec.push_back(0);
vec.push_back(1);
vec.push_back(2);
vec.push_back(3);
vec.push_back(4);
cout << vec.size() << endl;//5
cout << vec.capacity() << endl;//8
return 0;
}
PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL
apnacollege@Shradha DSAseries % g++ -std=c++11 code.cpp && ./a.out
3
4
apnacollege@Shradha DSAseries % g++ -std=c++11 code.cpp && ./a.out
5
8
apnacollege@Shradha DSAseries %
APNA COLLEGE

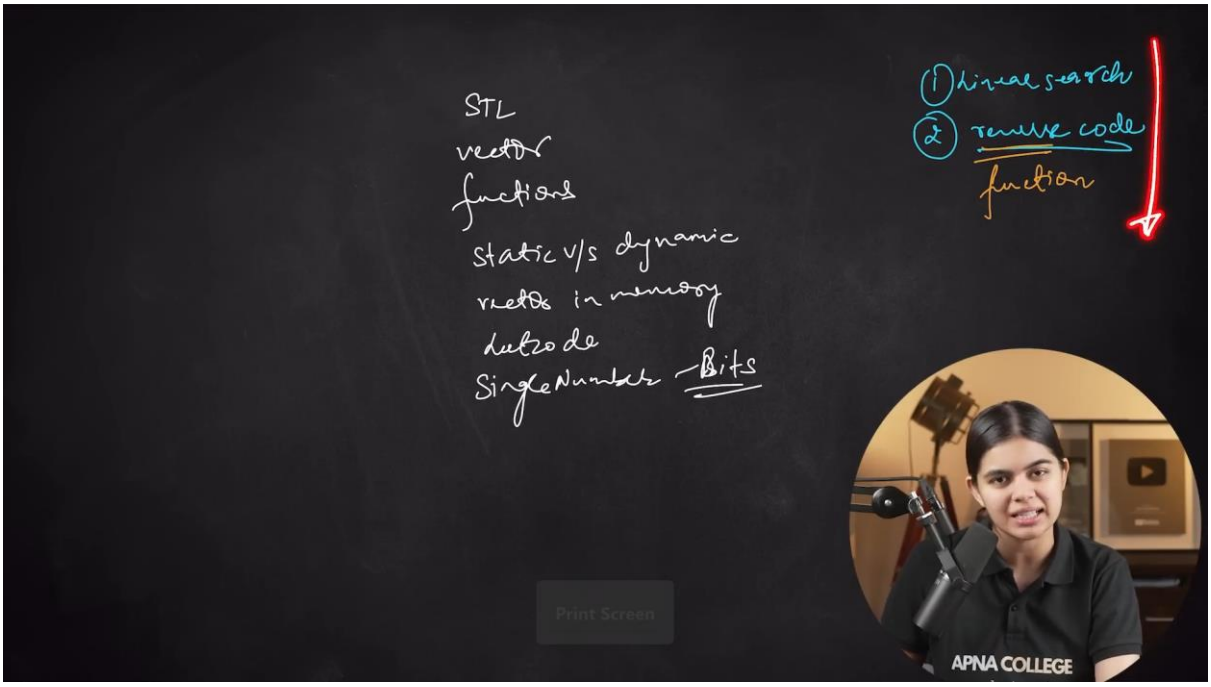

Single Number - LeetCode
leetcode.com/problems/single-number/
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
136. Single Number
Solved
Easy
Topics
Companies
Hint
Given a non-empty array of integers nums, every element appears twice except for one. Find that single one.
You must implement a solution with a linear runtime complexity and use only constant extra space.
Example 1:
Input: nums = [2,2,1]
Output: 1
Example 2:
Input: nums = [4,1,2,1,2]
Output: 4
Example 3:
Input: nums = [1]
Output: 1
Constraints:
1 <= nums.length <= 3 * 10^4
Code
C++
Auto
class Solution {
public:
int singleNumber(vector<int>& nums) {
}
};
Saved
Testcase
Test Result
Case 1
Case 2
Case 3
nums =
[2,2,1]
Print Screen
APNA COLLEGE


Single Number
BITWISE XOR
^
0^0 ⇒ 0
1^1 ⇒ 0
4 | 1 | 2 | 1 | 2
n^n = 0
4^1^2^1^2
4 ^ 0
n^n = 0
n^0 = n
Print Screen
APNA COLLEGE


Single Number
4 | 1 | 1
100
^ 001
101
^ 001
100
= 4
4 | 1 | 2 | 1 | 2
XOR ⇒ n
n^n = 0
n^0 = n
Print Screen
APNA COLLEGE


Single Number
4 1 2 1 2
XOR ⇒ n
ans = 0
for(int val : nums) {
ans = ans ^ val
}
return ans;
n ^ n = 0
n ^ 0 = n
APNA COLLEGE


Single Number - LeetCode
leetcode.com/problems/single-number/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
136. Single Number
Solved
Easy
Topics
Companies
Hint
Given a non-empty array of integers nums, every element appears twice except for one. Find that single one.
You must implement a solution with a linear runtime complexity and use only constant extra space.
Example 1:
Input: nums = [2,2,1]
Output: 1
Example 2:
Input: nums = [4,1,2,1,2]
Output: 4
Example 3:
Input: nums = [1]
Output: 1
Code
C++
Auto
class Solution {
public:
int singleNumber(vector<int>& nums) {
int ans = 0;
for(int val : nums) {
ans ^= val;
}
return ans;
}
};
Saved
Testcase
Test Result
Accepted
Runtime: 3 ms
Case 1
Case 2
Case 3
Input
nums =
APNA COLLEGE


STL
vector
functions
static v/s dynamic
vectors in memory
leetcode
SingleNumber - Bits
1) linear search
2) recursive code
function
APNA COLLEGE