Vectors, Casting, and Templates

Lesson #2 - 09/11/2025

Setup

Create a codeforces account at

https://codeforces.com

Once you have an account join the group by using the following url:

Also open the compiler to write code:

https://ide.usaco.guide

casting

We can use variables with other variables of the same type. This means that, for example, we cannot compare int and double directly. To do this, we need to use **casting**.

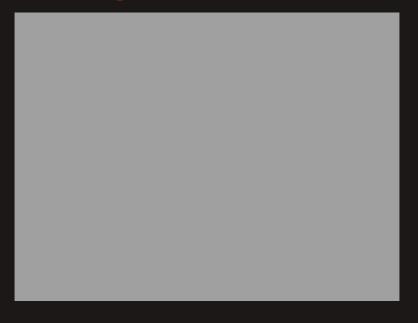
```
double decimal num = 4.69;
int number1 = decimal num;
cout << number1 << endl;</pre>
What is actually happening is the following:
double decimal num = 4.69;
int number1 = int(decimal num);
cout << number1 << endl;</pre>
```

casting - Continued

```
int big num = 500;
if(big num < 1e9){
}
if(double(big_num) < 1e9){</pre>
}
if(big num < (int)1e9){
```

Vectors

Vectors - Concept



Vectors - usage

Initializing vectors can be done like the following:

```
vector<T> my list; // starts out empty - contains no elements)
vector<T> my_list2(n); // starts with empty n elements
vector<T> my list3(n,default val); // starts with n-elements,
vector<int> num list:
vector<string> string list:
vector<double> double list; // etc...
vector<int> empty list(5);
vector<double> another list(5,3.14567);
vector<vector<int>> num list 2d(20, vector<int>(40, -5) );
```

Vectors - Looping through

```
vector<int> l = \{1.2.3.4.5.6.7\}:
for(int i = 0;i < l.size();i++){</pre>
  cout << l[i] << " ";
cout << endl; // prints the end line</pre>
for(int x : l){
  cout << x << " ":
}
cout << endl;</pre>
for(auto it = l.begin(); it != l.end();it++){
  cout << *it << " ":
cout << endl:</pre>
```

Vectors - Looping through (ii)

For 2d vectors similar syntax can be used:

```
int rows = 100;
int cols = 200:
vector<vector<int>>> dp(rows, vector<int>(cols, 0));
// this is for example how you can print a 2d list:
for(int row = 0; row < rows; row++){
  for(int col = 0;col < cols;col++){</pre>
    cout << dp[row][col] << " ";</pre>
  cout << endl;
```

Vectors - Methods

Vectors support lots of methods:

method	comment
<pre>push_back(T value)</pre>	add a value to the end of the list.
size()	Returns the number of elements in the vector.
front()	Returns the first element of the vector
back()	Returns the last element of the vector
erase(iterator)	Erases value at some index*
empty()	Returns true if empty, otherwise returns false.

```
vector<int> l = {10,20,30,50,20}; output:
l.push_back(-10); 5
l.erase(l.begin() + 2); 10 20 50 20 -10
cout << l.size() << endl;
for(int x : l)
    cout << x << " ";</pre>
```

Programming Problems' structure /

related topics

Reading in a list / lists

```
int n:
cin >> n: // read in number of elements
vector<int> nums;
for(int i = 0:i < n:i++){
  int tmp; cin >> tmp;
  nums.push back(tmp); // adds each element into the list
int n;
cin >> n; // read in number of elements
vector<int> nums(n);
for(int i = 0:i < n:i++){}
  cin >> nums[i];
```

Test cases

Input for each test case is two numbers:

Single test case	Multiple Test cases
2 3 // test case #1	3 // number of test cases 2 3 // test case #1 6 7 // text case #2 4 1 // test case #3

Input for each test case is a list:

Single test case	Multiple Test cases
5	2 // number of test cases
1 2 5 3 4	5 // start of test case 1
	1 2 5 3 4
	3 // start of test case 2
	1 2 3

Code Templates

Some code can often be annoying to type over and over, so we can make templates to make writing solutions easier

Code Template

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
using vi = vector<int>;
void solve(){
  ll long num = (ll)1e16;
  vi list of nums;
  vector<vector<int>> list 2d(10, vector<int>(40, 30) );
  vector<vi> list 2d(10, vi(40, 30));
int main(){
  int tc = 1:
  cin >> tc; // comment this out if its only one test case
}
```

Miscellaneous

- Practice competition
 - planning to have it this coming Tuesday
- · feedback on slides
 - ▶ theme
- next meeting
 - sets / hash maps
 - functions / recursion