

# Vector Indexing Solutions

- Give two different ways to loop over the elements of a vector

- Use an iterator

```
for (auto it = vec.begin(); it != vec.end(); ++it) {  
    cout << *it << "\t";  
}
```

- Use an index

```
for (int i = 0; i < vec.size(); ++i) {  
    cout << vec[i] << "\t";  
}
```

- Explain what is meant by indexing in the context of `std::vector`
  - An index is a number which indicates how far an element is from the start of the vector's memory block
    - e.g. `vec[0]` is at the start of the block (the first element)
    - `vec[1]` is one element away from the start of the block (the second element)
    - `vec[2]` is two elements away from the start of the block (the third element)

- Convert the code below into a full working program. Experiment with printing and assigning different elements until you are confident with it

```
vector<int> vec {4, 2, 3, 5, 1};  
cout << vec[0];  
cout << vec[3];  
vec[2] = 6;
```

- Add a loop to your program to print out all the values in the vector after changing the third element
- What happens if you try to print out `v[5]`? Explain your result

On my system, `vec[5]` printed out as 0

- What happens if you try to assign to `v[6]`? Explain your result

When I added the `vec[6]` assignment, all the output after `vec[3]` disappeared, indicating that the program had silently crashed

- What is the index of the first element in a vector?
  - 0
- Explain how to find the index of the last element in a vector
  - It will be one less than the number of elements
  - It can be computed as `vec.size() - 1`
- Why is it bad to access elements after this final index?
  - May be uninitialized (its value is garbage)
  - May be unallocated (causes memory access error)

- Convert the code below into a full working program
- Explain why the loop condition in the code below is

```
for (int i = 0; i < vec.size(); ++i) {
```

- The last element has index `vec.size()-1`
- When `i` is incremented after processing this element, `i` will be equal to `vec.size()`
- The conditional is then false
- The loop will then terminate, as required

- Explain how the code assigns a vector of even numbers
  - On each iteration, the index is 0, 1, 2, ...
  - The element is assigned the value  $2 \times \text{index}$
  - The elements will be 0,  $2 \times 1$ ,  $2 \times 2$ , ...
- Change this so the values are 2, 4, 6... instead of 0, 2, 4...



- Convert the code below into a full working program
- Explain how the code sample below prints out every second element
  - The index is increased by 2 on each iteration
  - $i = 0, 2, 4, \dots$
  - The elements accessed are `vec[0]`, `vec[2]`, `vec[4]`, ...

- Rewrite the for loop as a while loop

```
int i = 0;
while (i < vec.size()) {
    cout << vec[i];
    i += 2;
}
```

- Change your program from the previous slide to use the following string definitions instead of hello {'H', 'e', 'l', 'l', 'o'};
- Before running the program each time, try to guess what the output will be
  - string alpha {"abcdef"} // Elements are a, b, c, b, e, f
  - string zeds(6, 'z'); // Elements are z, z, z, b, z, z