New topic: vectors.

Motivational example: print stdin in reverse order.

fileA
```
aaa
bbb
ccc
ddd
```

```
./a.out < fileA
ddd
ccc
bbb
aaa
```

```
echo {1..10} | ./a.out
10
9
8
:
:
1
```

Issue: we don't know in advance how many variables we'll need. Vectors give one solution.

$V = \boxed{V[0] \,|\, V[1] \,|\, V[2] \,|\, V[3] \,|\, V[4]}$

How to add new elements? use "push_back(...)":

Say $V =$

`V.push_back(10);`	$V = \boxed{10}$
`V.push_back(20);`	$V = \boxed{10}\boxed{20}$
`V[1] = 9;`	$V = \boxed{10}\boxed{9}$

$\boxed{9}$

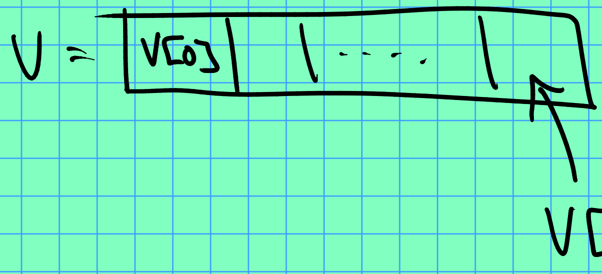
Question: what happens if I do `V[37] = 9;` ?

At the moment, there is no V[37]!!

⟹ Bad things will happen. Unpredictable behavior, or often a "segmentation violation"/ "seg fault".

have to use push_back(...), or resize(...)

Note: vectors know their own size: get it via V.size().



V[0, ... V.size()-1]

if V empty: V[0, ..., -1] (No elements)