EE 200 Lecture 6: Arrays

Steven Bell

21 September 2023

Tufts UNIVERSITY

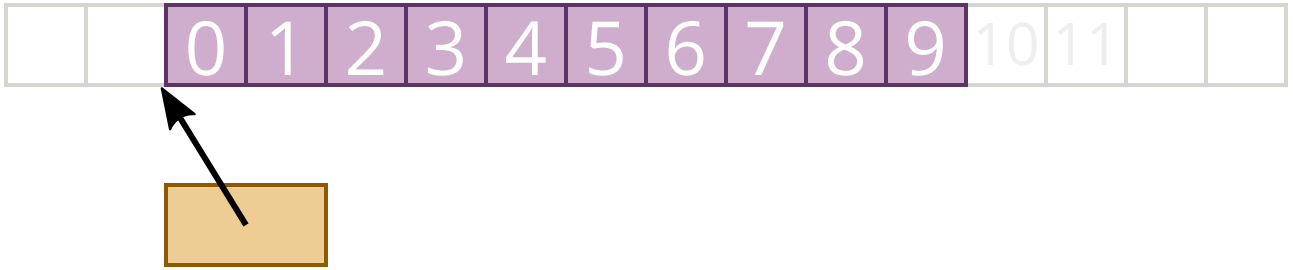
Indexing arrays

```
int grape[10]
```


```
for(int i = 0; i < 10; i++){
  grape[i] = 0;
}
```

Indexing arrays

```
int grape[10]
```




```
int* raisin = grape;
```

```
grape[0]
grape[10]
*(grape + 1)
*(raisin + 1)
```

```
raisin[1]

int i = 1;
i[grape];
```

sizeof

sizeof.c

With these declarations:

```
char pear[100];
char* pear_p;
```

```
sizeof pear[1]

sizeof pear

sizeof *pear_p

sizeof pear_p
```

```
int howBig(char arr[])
{
  return sizeof arr;
}
```

```
howBig(pear)

howBig(pear_p)
```

sizeof

sizeof.c

With these declarations:

```
char pear[100];
char* pear_p;
```

```
sizeof pear[1]

sizeof pear

sizeof *pear_p

sizeof pear_p
```

GCC will warn you about this:

```
int howBig(char arr[])
{
  return sizeof arr;
}
```

```
howBig(pear)

howBig(pear_p)
```

sizeof an array

Unlike Python, MATLAB, Java, etc., C doesn't track how big its arrays are.

So once you're outside the unit where it was declared, you have to track the size yourself.

Dangling pointers

dangle.c

Any pointer whose memory has been deallocated

Why is this bad?

A) You're reading data that might have changed

B) You're writing to memory that might be used for something else

C) The behavior is unpredictable

D) All of the above

Dangling pointers

For now, you can avoid dangling pointers by never returning a pointer to a local variable.

Once we can dynamically allocate memory, it will get more complex.

Make

Use "." to redo an action

Use visual mode to make edits to many lines at once:

Ctrl+v

navigate with h/j/k/l

Shift-i

make edit

Escape

now you're back in normal mode

Use `:s/one/two/g` for quick find-and-replace

Make

Make is a tool for automating compilation

Homework 5 is hosted on Github

Homework 3 resubmissions are due Tuesday 9/26 at 4:30pm

Homework 4 due Tuesday 9/26 at 4:30pm

Homework 5 due Thursday 9/28 at 4:30pm