Day 5 (Wed 2/2)

- Exercise 4 review
- Day 5 recap Qs
- arrays, C strings
- Exercise 5

HW0 due Friday 2/4
HW1 due Friday 2/11

slido.com
jhuintprog01

zero

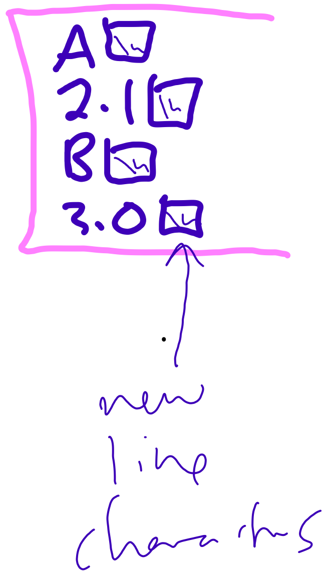
If you are remote:
Use slack workspace to ask for help during exercise

Exercise 4

- scanf returns the number of data values successfully read (so, useful for detecting end of input)

possible main loop

```
char grade;
float points;
int num_read = scanf("␣%c␣%f", &grade, &points);
while (num_read == 2) {
    handle grade
    num_read = scanf("␣%c␣%f", &grade, &points);
}
```



You usually want a space before %c when using scanf – tells scanf to skip whitespace characters

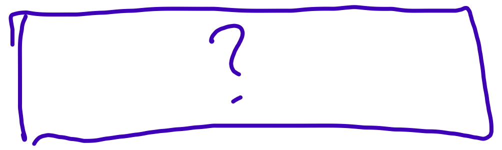
Exercise 4 (continued)

```
switch (grade) {
case 'A':
case 'a':
        code
    break;
  case 'B':
  case 'b':
        code
     break;

       :

  default:
        code if no case
        label was matched
     break;
}
```

must be careful not to forget a break statement at the end of the code that handles a particular case or cases

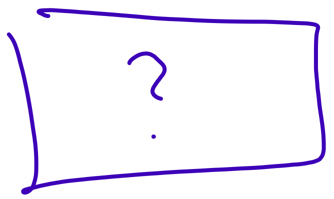
1. When we declare an array in C, what are the initial values?
2. What is the ASCII (Unicode) table?
3. What is a null terminator? What is its ASCII value?
4. Consider c-string "ab\0cd\0" - what is the reported string length?
5. How do we check if two C-strings are the same? In addition, are these two strings the same: "ab\0cd\0" and "ab\0"?

```
int a[4];

printf("%d\n", a[0]);
```



```
int x;

printf("%d\n",
        x);
```

use of
unitialized value



```
int a[4] = {1, 2, 3, 4};
int a[4] = {0}; // all elements will be zero
```

ASCII

char	integer	
A	65	
Ø	48	
␣	32	
\n	10	
\Ø	0	(NUL)

Unicode

ASCII
+
all other text characters

```
char s[10] = {'c', 's', '2', '2', '0', '\0'};
printf("%s\n", s);
```

```
CS220
```

```
char s[] = "ab\0cd\0";
printf("%d\n", (int)strlen(s));
```

$\boxed{2}$

```
char s1[] = "ab\0cd\0",
     s2[] = "ab\0";

int x = strcmp(s1, s2);
printf("%d\n", x);
```

$\boxed{0}$

```
char  str[1000];   // could store any string
                   // up to 999 chars
```

```
gcc -o myexe
```

```
char  a[10];

  ⋮

strcpy(a, "Hello, how are you?");
```

$\uparrow$

too small! $\Rightarrow$ undefined behavior

<u>valgrind</u>