PA1 Grades

- Resubmit assignment due next Thu

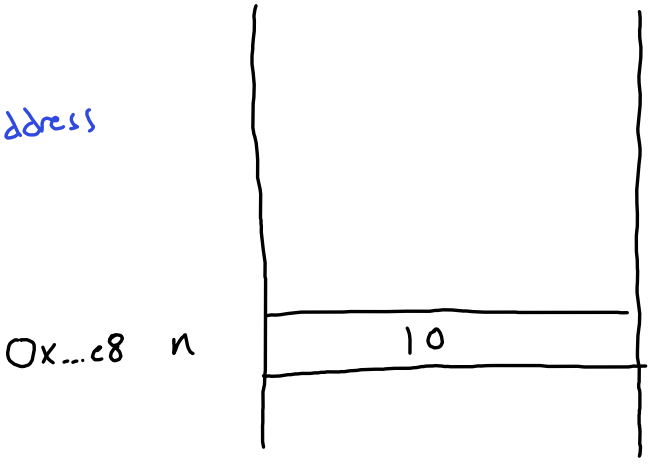
`&v` where v is some variable

This returns the address (so a pointer) where v is stored

```
int main() {
    int n = 10;
    &n
}
```

=> evaluate to the address 0x....c8

```
int* n_addr = &n;
```



Note: ran on Joe's Mac (long may differ)

%lu means "long unsigned"

```
1  #include <stdio.h>
2  #include <stdint.h>
3
4  void wheresmystuff(char* s) {
5    int x = 12;
6    uint8_t y = 53;
7    char z = 9;
8    double ns[] = { 4.0, 3.0, 9.0 };
9
10   printf("x=%d, %lu bytes, starts at: %p\n", x, sizeof(x), &x);
11   printf("y=%hhu, %lu bytes, starts at: %p\n", y, sizeof(y), &y);
12   printf("z=%hhd, %lu bytes, starts at: %p\n", z, sizeof(z), &z);
13   printf("ns=[%f,%f,%f], %lu bytes, starts at: %p\n",
14          ns[0], ns[1], ns[2], sizeof(ns), &ns);
15
16   printf("s=\"%s\"@%p, %lu bytes, starts at: %p\n", s, s,
         sizeof(s), &s);
17 }
18
19 int main() {
20   char str[] = "14 char string";
21   wheresmystuff(str);
22   printf("\nstr takes up %lu bytes starting at: %p\n",
         sizeof(str), &str);
23 }
```

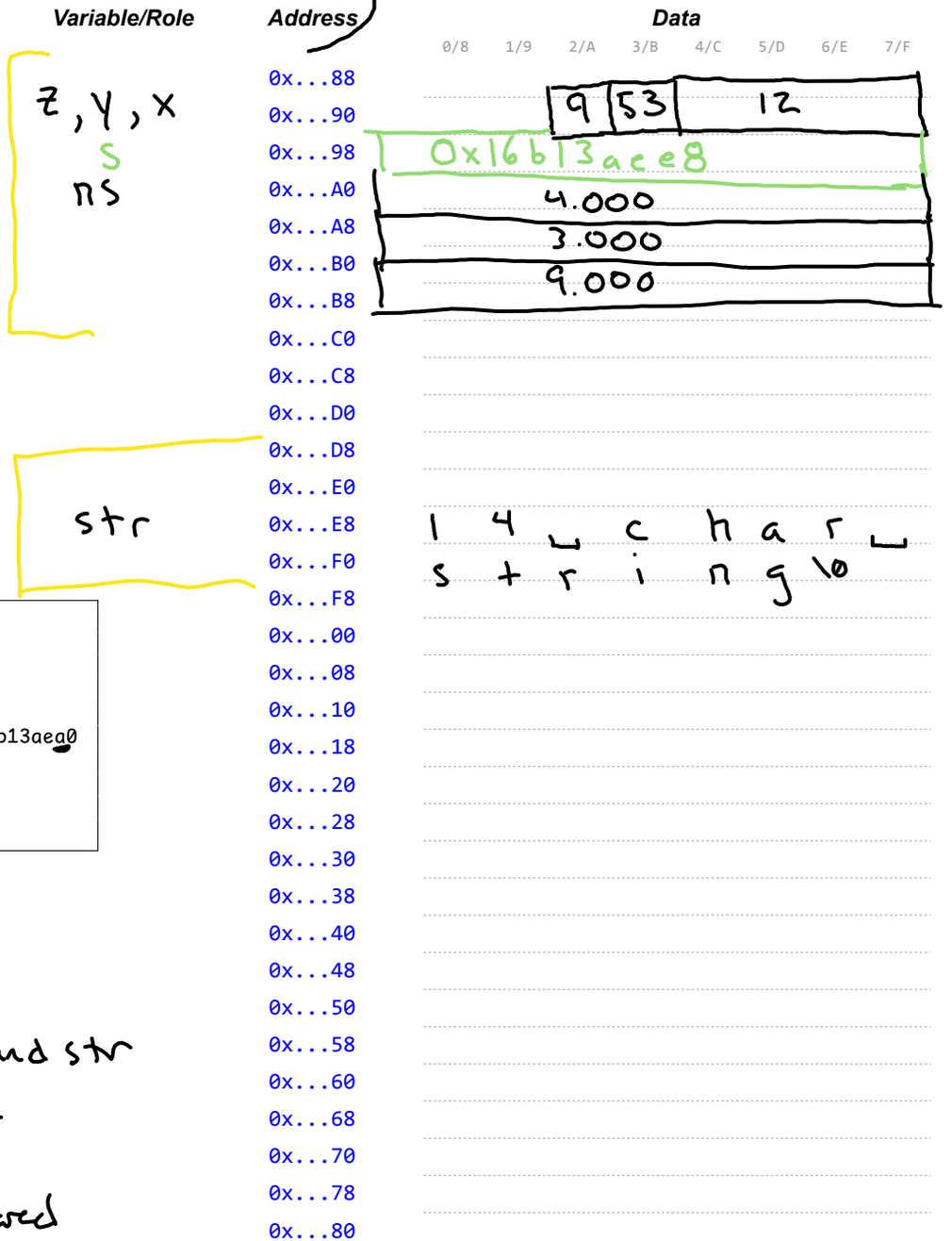
```
$ gcc -Wall wheresmystuff.c -o wheresmystuff
$ ./wheresmystuff
x=12 takes up 4 bytes starting at: 0x16b13ae94
y=53 takes up 1 bytes starting at: 0x16b13ae93
z=9 takes up 1 bytes starting at: 0x16b13ae92
ns=[4.000000,3.000000,9.000000] takes up 24 bytes starting at: 0x16b13aea0

s="14 char string"@0x16[redacted]e8, 8 bytes, starts at: 0x16[redacted]98

str takes up 15 bytes starting at: 0x16b13aee8
```

stack frame wheresmystuff

stack frame for main



&s

Really the data stored in variable s

There is a difference between s and str

- s is a pointer/address argument
- str is an array variable declared in main

sizeof(v) tells us the space need for that variable's value in this function