

Quick Review/Refresh: sizeof can take types, too!

$$\text{sizeof}(\text{uint8_t}) = \underline{1}$$

$$\text{sizeof}(\text{char}^*) = \underline{8}$$

$$\text{sizeof}(\text{uint32_t}^*) = \underline{8}$$

$$\text{sizeof}(\text{uint32_t}) = \underline{4}$$

$$\text{sizeof}(\text{char}) = \underline{1}$$

```

1 #include <stdio.h>
2 #include <string.h>
3
4 char* concat(char a[], char b[]) {
5     int alen = strlen(a), blen = strlen(b);
6     char result[alen + blen + 1];
7     for(int i = 0; i < alen; i += 1) {
8         result[i] = a[i];
9     }
10    for(int i = 0; i < blen; i += 1) {
11        result[i + alen] = b[i];
12    }
13    result[alen + blen] = 0;
14    char* returnPtr = result;
15    return returnPtr;
16 }
17
18 int main() {
19     char like[] = "I like ";
20     char c1[] = "apples";
21     char c2[] = "CSE29";
22
23     printf("like: %p c1: %p c2: %p\n", like, c1, c2);
24
25     char* result1 = concat(like, c1);
26     char* result2 = concat(like, c2);
27     printf("%p %s\n", result1, result1);
28     printf("%p %s\n", result1, result2);
29
30 }

```

same printing

```

$ gcc -Wall concat-ptr.c -o concat-ptr
$ ./concat-ptr
like: 0x7ffc215ae970 c1: 0x7ffc215ae969 c2: 0x7ffc215ae963
0x7ffc215ae8e0 I like CSE29
0x7ffc215ae8e0 I like CSE29

```

Variable/Role

Address

Data

0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

0x...88
0x...90
0x...98
0x...A0
0x...A8
0x...B0
0x...B8
0x...C0
0x...C8
0x...D0
0x...D8
0x...E0
0x...E8
0x...F0
0x...F8
0x...00
0x...08
0x...10
0x...18
0x...20
0x...28
0x...30
0x...38
0x...40
0x...48
0x...50
0x...58
0x...60
0x...68
0x...70
0x...78
0x...80

(You may hear abt malloc - not even once)

Should not return address allocated on the stack

Instead, take a result param)

concat

a
b
result

0x...970
~~0x...969~~ 0x...963
I l i k e w a c
s p p l e s 10
e 2 9 10

result2
result1

main

0x...8e0 (stored @ line 26)
0x...8e0 (stored @ line 25)

C S E 2 9
10 a p p l e s 10
I l i k e 10

Why not $\&\text{result}$ on line 14?

Observation: 2 concat calls return same address

For an array variable a
stored in a pointer var
~~a~~ or printed with
 $\%p$, $a == \&a$

Should print "I like apples" first

Could we return result directly?

If we do return result on line 14,
compiler gives a warning!

Warning: returning address of a stack-allocated
array.

Can I do

$\text{char returnPtr}[] = \underline{\text{result}};$ X syntax error

This cannot be any
expression.

```
$ ls -l pa2  
$ cd  
$ git commit ...
```

} bash
≡
shell

```
while (1) {  
    printf("$ ");  
    fgets(.....)  
    // runs that command  
}
```