

CS107 Lecture 13: Two-Minute Etudes

Problem 1: Generics and `void *` Pitfalls

For each of the following programs, present the expected output to the right. If you think the program might crash or exhibit undefined behavior, then say so and explain why. These programs rely on the generic implementation of `swap` and `swap_ends`, presented right [here](#) and [here](#). Assume the `print_int_array` function does precisely what its name implies.

You should not write code like this. 😊

- a.
- ```
int x = 6, y = 12;
swap(&x, &y, sizeof(short));
printf("%d %d\n", x, y);
```
- b.
- ```
char word[] = "gateway";
swap(word + 1, word + 3, sizeof(char));
printf("%s\n", word);
```
- c.
- ```
char *s1 = "approachable";
char *s2 = "considering";
swap(&s1, &s2, sizeof(char *));
printf("%s %s\n", s1, s2);
```
- d.
- ```
char *s1 = strdup("approachable");
char *s2 = strdup("considering");
swap(s1, s2, sizeof(char *));
printf("%s %s\n", s1, s2);
free(s1);
free(s2);
```
- e.
- ```
int array[] = {1, 2, 3, 4, 5, 6, 7, 8};
swap_ends(array, 8, sizeof(int));
print_int_array(array, 8);
```
- f.
- ```
int array[] = {1, 2, 3, 4, 5, 6, 7, 8};
swap_ends(array + 1, 3, sizeof(char *));
print_int_array(array, 8);
```

Answers:
a. 12 6 on myths, 6 12 if big-endian.
b. gateway
c. considering approachable
d. considerable approaching
e. 8 2 3 4 5 6 7 1
f. 1 6 7 4 5 2 3 8

Problem 2: Manual Pointer Arithmetic

For each of the following programming snippets, present the expected output to the right. If you think the program might crash **or exhibit undefined behavior**, then say so and explain why. Assume **myth**-machine behavior, since these tricks aren't necessarily portable.

a.

```
char alphabet[] = "abcdefghijklmnop";
char *inset = (char *)((int *) alphabet + 2); // yes, you can do this
printf("%s\n", inset);
```

b.

```
short *ptr = malloc(2 * sizeof(short));
ptr[0] = ptr[1] = 1;
printf("%d\n", *(int *)ptr); // answers of the form x * 2^16 + y are fine
free(ptr);
```

c.

```
int primes[] = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29};
long *lucky = (long *)&primes[3] + 1;
printf("%ld\n", lucky[2]); // answers of the form x * 2^32 + y are fine
```

d.

```
const char *stranger[] = { "Mike", "Dustin", "Lucas", "Will", "Max" };
char things[20];
for (size_t i = 0; i < 5; i++) {
    memcpy(&((int *)things)[i], stranger[i], 4);
}
printf("%s\n", &things[11]);
```

Answers:
a.abcdefghijklmnop
b.65537
c.124554051607 on the myths and other
little-endian machines, 98784247837 on big-
endian machines
d.awillmax