

CS107 Lecture 12: Two-Minute Etudes

Problem 1: malloc Mayhem and realloc Realness

For each snippet, give the output before the first memory error and identify and explain that error. If none occurs, give the full output, noting any leaks. Assume all allocations succeed, and valid calls to **realloc** with non-**NULL** pointers always move the allocation to a new address.

a.

```
char *lake = malloc(6);
strcat(lake, "water");
printf("%s\n", lake);
free(lake);
```

b.

```
char *crepe = strdup("flapjack");
char *pancake = crepe + 4;
printf("%s\n", pancake);
pancake = realloc(pancake, 20);
free(pancake);
```

c.

```
char *caterpillar = strdup("caterpillar");
char *littlefeet = caterpillar + strlen(caterpillar);
caterpillar = realloc(caterpillar, 30);
strcpy(littlefeet, " -> butterfly");
printf("%s\n", caterpillar);
free(caterpillar);
```

d.

```
char *cookie = NULL;
cookie = realloc(cookie, 12);
strcpy(cookie, "bis");
strcat(cookie, "cuit");
printf("%s\n", cookie);
free(cookie);
```

e.

```
char *approval = strdup("countenance");
approval[5] = '\0';
approval = realloc(approval, 6);
printf("%s\n", approval);
free(approval);
```

Answers:
a. No output; **strcat** reads uninitialized memory.
b. Prints **jack**, then passes an invalid interior pointer to **realloc**.
c. No output; **strcpy** writes through the stale **littlefeet** pointer.
d. Prints **biscuit**, with no errors or memory leaks.
e. Prints **count** without error or memory leaks.