CS 3: Introduction to Software Design

Pointers Exercises

Name: □

Warmup

fread Prototype Reminder

```
size_t fread(void *ptr, size_t size, size_t nitems, FILE *stream);
```

Fill In The Blanks!

```
void read_one_1() {
    □ c = malloc(sizeof(char));
    fread(□, sizeof(char), 1, stdin);
    printf("I got: %c\n", □);
}
```

```
void read_one_2() {
    □ c = 'X';
    fread(□, sizeof(char), 1, stdin);
    printf("I got: %c\n", □);
}
```

More &

```
int to_nibble(char *bin) {
    char *endptr = NULL;
    char *dup = strndup(bin, 4);
    int result = strtol(dup, □, 2)
      ;
    if (□ != endptr) {
        free(dup);
        return -1;
    }
    free(dup);
    return result;
}
```

```
long strtol(char *str, char **endptr, int base)
------------------------------------------------
If endptr is not NULL, strtol() stores the
address of the first invalid character in
*endptr.  If there were no digits at all,
however, strtol() stores the original value of
str in *endptr.  (Thus, if *str is not `\0' but
**endptr is `\0' on return, the entire string
was valid.)
```

Pointer Arithmetic

```c
int strip_leading_zeroes(char **ptr) {
    int result = 0;
    while ((*ptr)[0] == '0' && (*ptr)[1] != '\0') {
        

        
    }
    return result;
}
```

All Together Now...

```c
char *readline(char **buf) {
    int charsread = 0;
    char c = '\n';
    while (fread(          , sizeof(char), 1, stdin) && c != '\n') {
        charsread++;
        

        
    }
    return 
}
```

Bonus

```c
char *pad_to_n(char *str, size_t n) {
    size_t len = (strlen(str) / n + 1) * n;
    char *out = calloc(len + 1, sizeof(char));
    char *ptr = out;
    for (size_t i = 0; i < len - strlen(str); i++) {
        *ptr = '0';
        ptr++;
    }
    while (          ) {
        
        ptr++;
        str++;
    }
    return out;
}
```