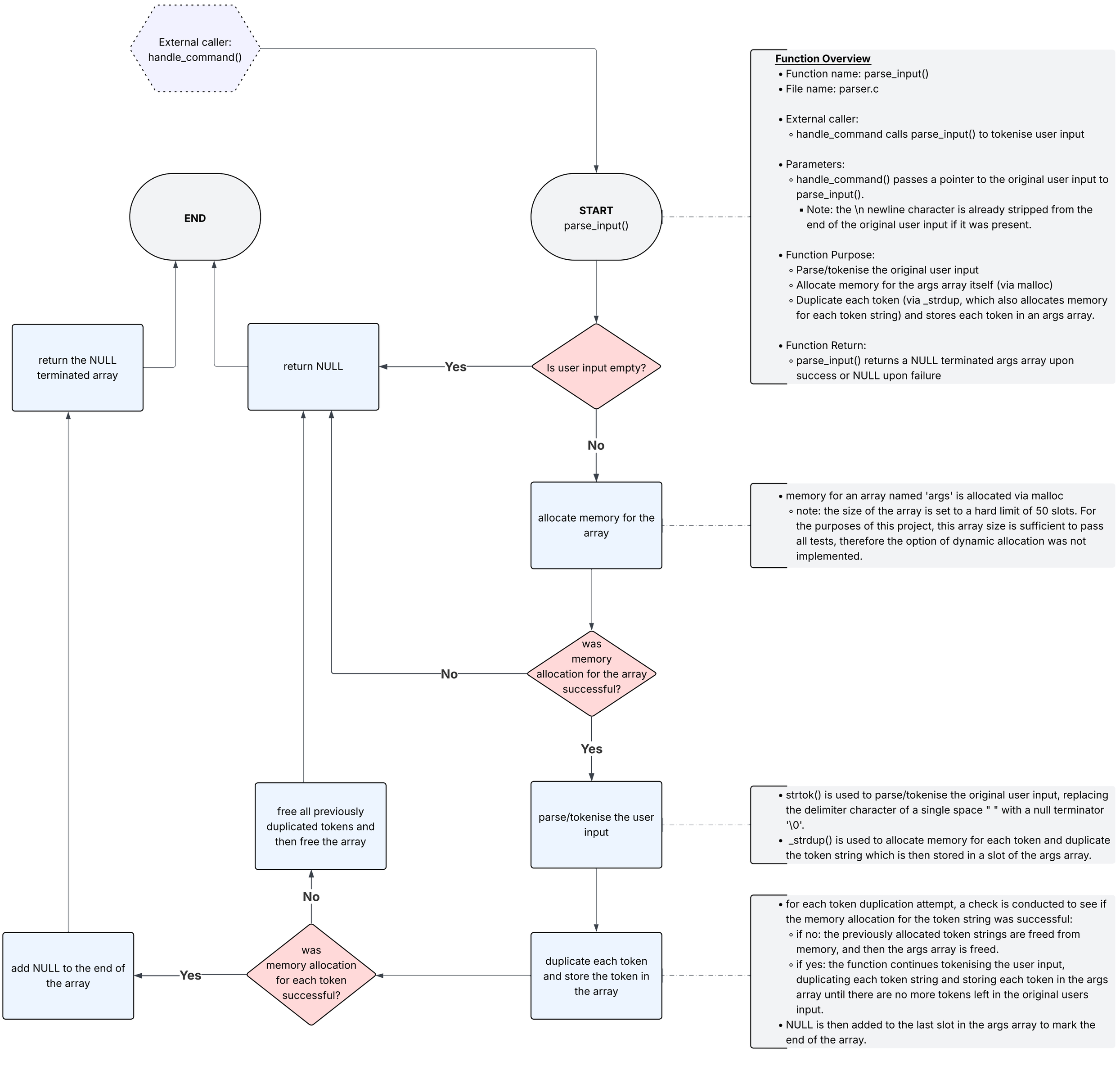
Holberton Simple Shell
High Level Design

parse_input() flowchart



Function Overview

- Function name: parse_input()
- File name: parser.c

- External caller:
  - handle_command calls parse_input() to tokenise user input

- Parameters:
  - handle_command() passes a pointer to the original user input to parse_input().
    - Note: the \n newline character is already stripped from the end of the original user input if it was present.

- Function Purpose:
  - Parse/tokenise the original user input
  - Allocate memory for the args array itself (via malloc)
  - Duplicate each token (via _strdup, which also allocates memory for each token string) and stores each token in an args array.

- Function Return:
  - parse_input() returns a NULL terminated args array upon success or NULL upon failure

- memory for an array named 'args' is allocated via malloc
  - note: the size of the array is set to a hard limit of 50 slots. For the purposes of this project, this array size is sufficient to pass all tests, therefore the option of dynamic allocation was not implemented.

- strtok() is used to parse/tokenise the original user input, replacing the delimiter character of a single space " " with a null terminator '\0'.
- _strdup() is used to allocate memory for each token and duplicate the token string which is then stored in a slot of the args array.

- for each token duplication attempt, a check is conducted to see if the memory allocation for the token string was successful:
  - if no: the previously allocated token strings are freed from memory, and then the args array is freed.
  - if yes: the function continues tokenising the user input, duplicating each token string and storing each token in the args array until there are no more tokens left in the original users input.
- NULL is then added to the last slot in the args array to mark the end of the array.