

struct \*

struct



# comp1511 week 7

starting ~9:05am

# notices

- assignment 0 marks have been released
  - submissions tab -> assignment 0 -> click blue dot for more details
- congratulations for finishing assignment 1!!!
  - will be marked by mid-week 9
  - I'll try to provide general style feedback next week in the tute

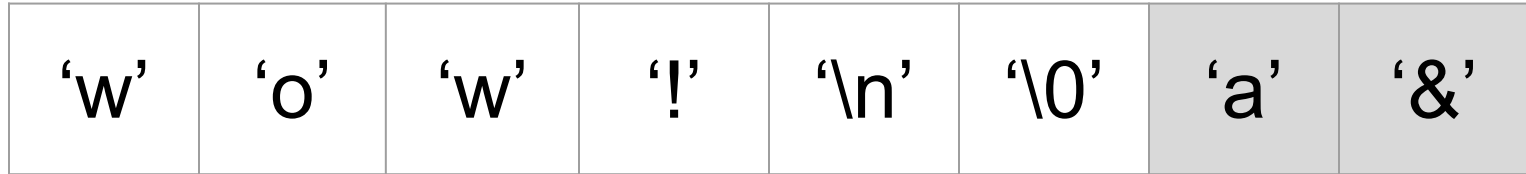
## ✨ reflection time ✨

*if you could go back in time and give one piece of advice to your past self, what would you say?*

# today

- strings
- fgets
- struct pointers

## strings



↑  
**NULL**  
terminator

```
int secret_function(char word[SIZE]) {  
    int i = 0;  
    int result = 0;  
    while (word[i] != '\0') {  
        if (word[i] >= 'a' && word[i] <= 'z') {  
            result++;  
        }  
        i++;  
    }  
    return result;  
}
```

- what does the function do?
- why is `\0` the terminating condition of the while loop?
- what is the difference between `char word[]` and `char *word`

# fgets

how do we use fgets?

what inputs are given to fgets?

how does fgets signal it has reached the end of its input?

**try:** man 3 fgets (in terminal)

## Description

The C library function `char *fgets(char *str, int n, FILE *stream)` reads a line from the specified stream and stores it into the string pointed to by **str**. It stops when either **(n-1)** characters are read, the newline character is read, or the end-of-file is reached, whichever comes first.

## Declaration

Following is the declaration for fgets() function.

```
char *fgets(char *str, int n, FILE *stream)
```

## Parameters

- **str** – This is the pointer to an array of chars where the string read is stored.
- **n** – This is the maximum number of characters to be read (including the final null-character). Usually, the length of the array passed as str is used.
- **stream** – This is the pointer to a FILE object that identifies the stream where characters are read from.

## Return Value

On success, the function returns the same str parameter. If the End-of-File is encountered and no characters have been read, the contents of str remain unchanged and a null pointer is returned.

If an error occurs, a null pointer is returned.

[https://www.tutorialspoint.com/c\\_standard\\_library/c\\_function\\_fgets.htm](https://www.tutorialspoint.com/c_standard_library/c_function_fgets.htm)

# fgets: how it works

```
char *fgets(char *str, int n, FILE *stream)
```

1. **stream**: will be ``stdin`` in 1511 (keyboard)
  - other examples include files
2. type to your heart's content up to **size n**
  - if you type more, it'll basically be ignored
3. hit **enter** which will trigger the ``fgets`` command
4. fgets stores what's in stdin up until the **newline** in **char \*str**
  - or at (n-1) characters, or end-of-file is reached, whichever first

**code demo:** echo\_twice.c



# struct pointers

Below is a struct definition for a student which will be used for the next set of questions.

```
struct student {  
    int zID;  
    double wam;  
    char name[MAX_NAME_LENGTH];  
};
```

12. How would you create a variable, `stu`, which is a struct `student`?
13. How would you create a variable, `stu_pointer`, that points to this new struct?
14. How would you give `stu` the following values by **only using this new pointer**?
  - `zID`: 5123456
  - `wam`: 74.7
  - `name`: Frankie
15. What is the use of the `->` operator? Change the previous code to utilise it.

# debugging 101

```
#include <stdio.h>

int main(void) {
    char string* = "";
    char c;
    scanf("%c", &c);
    int chars_read = 0;
    while (c != eof) {
        string[chars_read] = c;
        scanf("%c", &c);
    }
    printf("Read in %d chars for the string: %s", chars_read, string);
    return 0;
}
```