▷ <u>Reading Line by Line</u>

fgets (buffer, size, stream)
- keeps extracting characters from <u>STREAM</u> and puts them into <u>BUFFER</u> until :

     1. end of file becomes true
     2. the newline character is expected and stored in
         BUFFER
     3. SIZE - 1 characters have been stored in <u>BUFFER</u>

     A NULL CHARACTER is then APPENDED
     - returns the null pointer on read error or end-of-file
       with no character extracted.
     - null pointer = <u>NULL</u> = $\emptyset$

<u>Standard Idiom to process stdin line by line</u>
```
#define linesize 1024    /* linesize is a macro w/ a
char line [LINESIZE];    larger int value */
while (fgets (line, LINESIZE, stdin))
    /* process line */
```

- **Example - Line by line**

```
char line [10];
fgets (line, 10, stdin);
printf ("%.s ", line);
fgets (line, 10, stdin);
printf ("%.s", line);
fgets (line, 10, stdin);
printf ("% s", line);
fgets (line, 10, stdin);
printf ("%s", line);
```

| ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

↓ hello ↵

| h | e | l | l | o | \n | \0 | ? | ? | ? | hello\n

↓ hi ↵

| h | i | \n | \0 | o | \n | \0 | ? | ? | ? | hi\n

↓ hello_world↵

| h | e | l | l | o | _ | w | o | r | \0 | hello wor

* ld from Hello world is in input buffer. doesn't wait for user to input data

| l | d | \n | \0 | o | _ | w | o | r | \0 |

Lesson
1. use a big buffer so we don't overload it
2. You don't need to clear buffer with _fgets_

▷ <u>Reading data item by data item</u>

scanf   read from stdin
{ fscanf   read from a file
{ sscanf   reads from a string

↳ both take an extra (first) argument that specifies file/string

Ex. int n;
     scanf ("%d", &n);
           ↑
       conversion specs
       similar to printf

Address of operator:
we are passing address
of n to scanf.

– in C, everything is passed by value. If we can pass n
into scanf, scanf gets a copy and won't be able to
change the ORIGINAL n. Here, scanf gets a copy of
address of n but copy allows us to get to original

char s [100];   /* Note: no &, array name is address*/
scanf ("%s, s);

%d	int in base 10	scanf skips loading whitespace
%s	Word	
%c	char	doesn't skip whitespace
%f	float } f is	
%Lf	double } used for	
	both	