Begin by Initializing the Stack

Let's write the code.

```
int main ()
{
    char buf[200];
    struct stack_t stack;

    stack.top = 500;
```

a buffer to store one line

a stack

Initialize stack to empty.

Read from Keyboard Until Stack Full or Input Ends

```
while (0 < stack.top &&
       NULL != fgets
           (buf, 200, stdin)) {
    strcpy (stack.data[--stack.top],
            buf);
}
```

Stack not full?

Got line from keyboard?

Copy from **buf** into stack.

Decrement, then use index.

Logical AND Shortcutting Prevents Read with Full Stack

```
while (0 < stack.top &&
       NULL != fgets
           (buf, 200, stdin)) {
    strcpy (stack.data[--stack.top],
            buf);
}
```

Important: If the stack is full, no line is requested (**fgets** is not called).

Print a Line, Pop, and Repeat Until Stack is Empty

```
while (500 > stack.top) {
    printf ("%s",
        stack.data[stack.top++]);
}
return 0;
} // end of main
```

Stack not empty?

Print one line (includes LF).

Use index, then increment.

