**Designing an Abstract Data Type (ADT)**

**data object: storage for the input buffer so that the last added is marked in some way**

operations on that data object:

**delete the last-added entry in the input buffer**

**add character to the input buffer as the last-added**

When you type a line of text on the keyboard

and make mistakes, what do you do?

If you type backspace which I will denote as the character ‘#’ since we cannot really “see’ the backspace character

abcc#ddde###ef#fg

abcdefg

What is really the end result?

Let's draft an algorithm in pseudo-code (English like)

to add and remove characters from a "storage structure" that represents the input buffer

// read a line of text, correcting mistakes

read a new character achar

while ( not at the end of the line)

{

if(it’s a backspace character and input buffer is not empty)

delete the last-added entry in the input buffer

else if id it’s a backspace character and buffer is empty

do nothing

else

add character to the input buffer as the last-added

read a new character achar

}

test the algorithm on **ab#c** and then on **a##**

ac

make a directory **LABSEPT29**