

give_type_prompt	
	<u>return start time</u>
	move input into \$tx
SR	print prompt
SR	print string
SR	syscall 30
	move lower 32 bits into \$v0 as output
	update stack every time sub routine (SR) is called
	push/pop \$ra

check_user_input_string	
	<u>return 0,1</u>
SR	compare_strings
	calculate time end - start
	compare to allotted time
	if user is within time limit and strings match return 1 else 0
	push/pop \$ra

compare_strings	
	<u>return 0,1</u>
	while loop:
	condition: while lb is not null
	load each byte from string and send as input to compare_chars
SR	compare_chars
	keep track of rolling sum using output of compare_chars
	return 1 if sum is 0 else return 0
	push/pop \$ra

compare_chars	
	<u>return 0, 1</u>
	conditional statement

EXTRA CREDIT	
	<u>nested labels</u>
	for case: make all letters lower case before sending to compare chars
	for punctuation: check if within range of punctuation, if within load next byte check if space because this is in range too

