check50

cs50/problems/2025/x/scrabble

:) scrabble.c exists

Log

```
checking that scrabble.c exists...
```

:) scrabble.c compiles

Log

```
running clang scrabble.c -o scrabble -std=c11 -ggdb -lm -lcs50...
```

:) handles letter cases correctly

Log

```
running ./scrabble...
sending input LETTERCASE...
sending input lettercase...
checking for output "Tie!"...
checking that program exited with status 0...
```

:) handles punctuation correctly

Log

```
running ./scrabble...
sending input Punctuation!?!?...
sending input punctuation...
checking for output "Tie!"...
checking that program exited with status 0...
```

:) correctly identifies 'Question?' and 'Question!' as a tie

Log

```
running ./scrabble...
sending input Question?...
sending input Question!...
```

```
checking for output "Tie!"...
checking that program exited with status 0...
```

:) correctly identifies 'drawing' and 'illustration' as a tie

Log

```
running ./scrabble...
sending input drawing...
sending input illustration...
checking for output "Tie!"...
checking that program exited with status 0...
```

:) correctly identifies 'hai!' as winner over 'Oh,'

Log

```
running ./scrabble...
sending input Oh,...
sending input hai!...
checking for output "Player 2 wins!"...
checking that program exited with status 0...
```

:) correctly identifies 'COMPUTER' as winner over 'science'

Log

```
running ./scrabble...
sending input COMPUTER...
sending input science...
checking for output "Player 1 wins!"...
checking that program exited with status 0...
```

:) correctly identifies 'Scrabble' as winner over 'wiNNeR'

Log

```
running ./scrabble...
sending input Scrabble...
sending input wiNNeR...
checking for output "Player 1 wins!"...
checking that program exited with status 0...
```

:) correctly identifies 'pig' as winner over 'dog'

Log

```
running ./scrabble...
sending input pig...
sending input dog...
checking for output "Player 1 wins!"...
checking that program exited with status 0...
```

:) correctly identifies 'Skating!' as winner over 'figure?'

Log

```
running ./scrabble...
sending input figure?...
sending input Skating!...
checking for output "Player 2 wins!"...
checking that program exited with status 0...
```

:) correctly identifies winner between random words

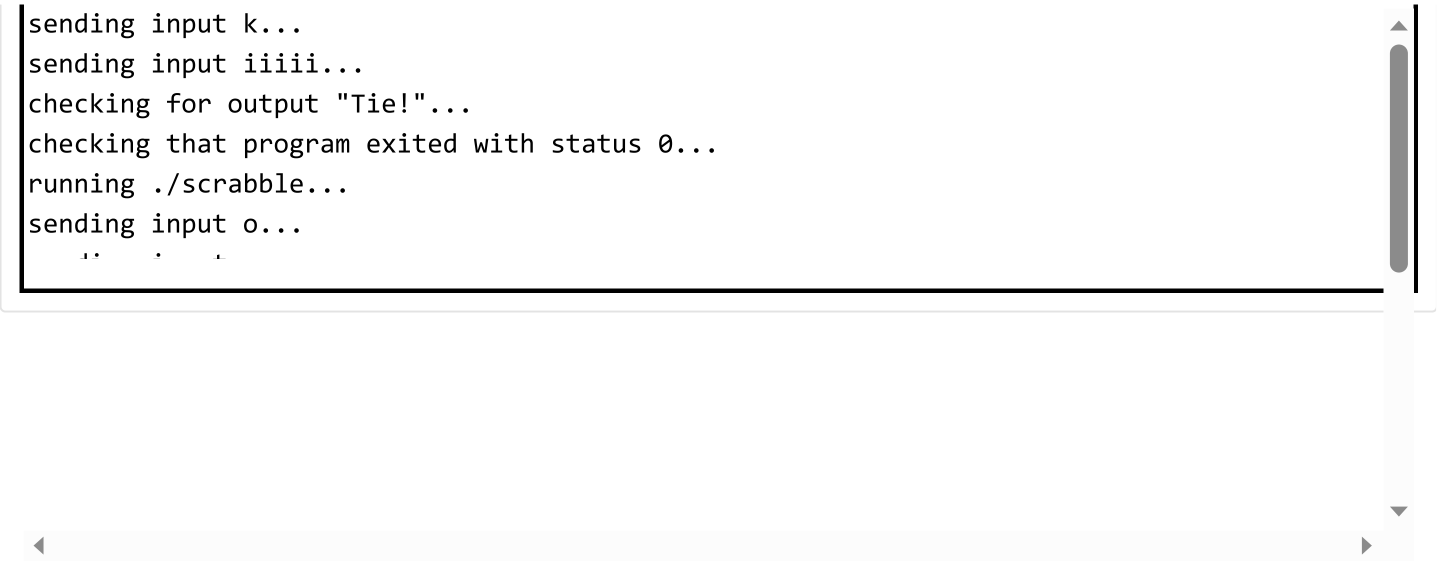
Log

```
running ./scrabble...
sending input h...
sending input i...
checking for output "Player 1 wins!"...
checking that program exited with status 0...
running ./scrabble...
sending input y...
sending input z...
checking for output "Player 2 wins!"...
checking that program exited with status 0...
running ./scrabble...
sending input c...
sending input d...
```

:) implementation correctly calculates scores using the Scrabble points table

Log

```
running ./scrabble...
sending input q...
sending input uuuuuuuuuu...
checking for output "Tie!"...
checking that program exited with status 0...
running ./scrabble...
```



```
sending input k...
sending input iiiii...
checking for output "Tie!"...
checking that program exited with status 0...
running ./scrabble...
sending input o...
```