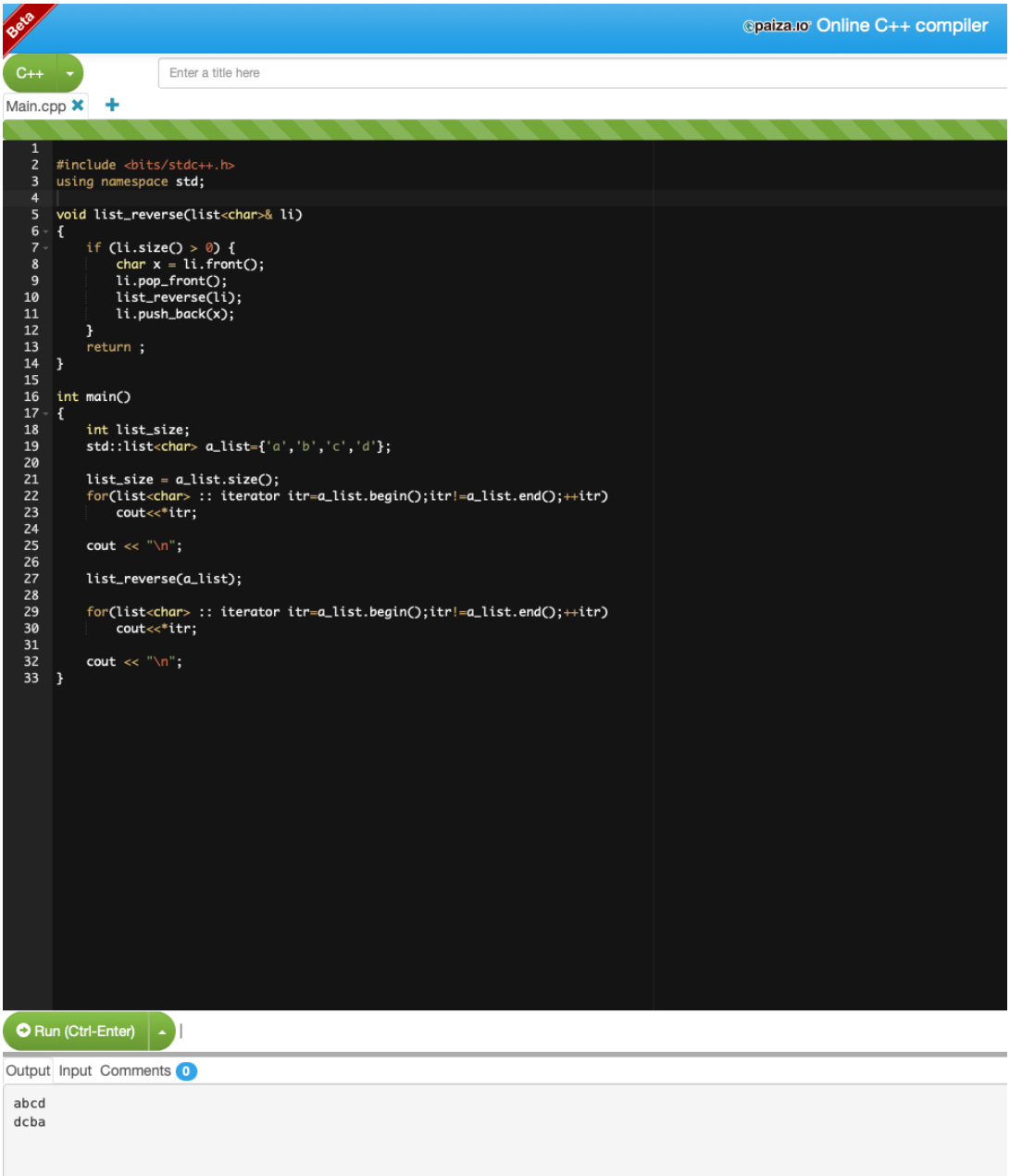
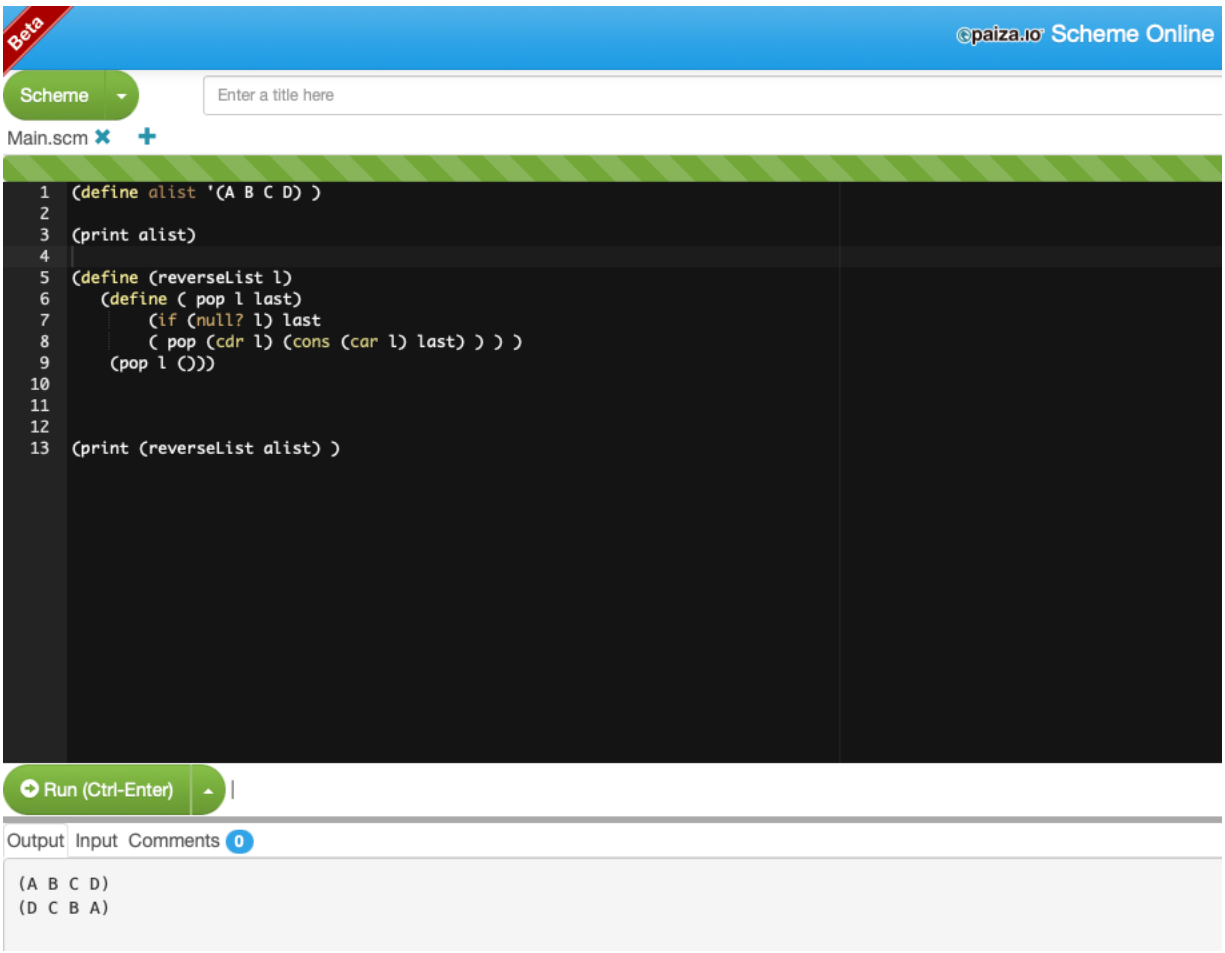
CS301
Programming Assignment 4
Thomas Crow
12/5/2022

C++



Scheme


Beta
paiza.io Scheme Online
Scheme
Enter a title here
Main.scm
(define alist '(A B C D))
(print alist)
(define (reverseList l)
(define (pop l last)
(if (null? l) last
(pop (cdr l) (cons (car l) last))))
(pop l ()))
(print (reverseList alist))
Run (Ctrl-Enter)
Output Input Comments 0
(A B C D)
(D C B A)

• Do you think Functional programming languages should be used instead of imperative languages?

There is no such thing as a one-size fit all programming language. (Despite what they told us about Java in the 1990s). In fact, many languages incorporate both paradigms into their language. There are some compelling benefits of functional languages:

- Program bugs are less likely due to no external side effects
- Efficient parallel programming
- Improved modularity
- Cleaner and more efficient nested functions

• Which of both programs is more efficient? Why do you think that?

Code-wise, the Scheme program is more efficient. The number of lines of code required to implement the recursive function was far fewer in Scheme. The recursive nature of the algorithm lends to the strength of functional programming.

• Do you think this type of language is easier to read than imperative languages?

While I believe that there may be a higher learning curve with functional programming, I believe that once used it, functional programming may have a significant advantage, especially when debugging someone else's code. As calling a function with a specific argument will always return the same result regardless of when it is run.