

CS301

Programming Assignment 4

Thomas Crow

12/5/2022

C++

Beta

paiza.io Online C++ compiler

C++ Enter a title here

Main.cpp

```
1
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 void list_reverse(list<char>& li)
6 {
7     if (li.size() > 0) {
8         char x = li.front();
9         li.pop_front();
10        list_reverse(li);
11        li.push_back(x);
12    }
13    return ;
14 }
15
16 int main()
17 {
18     int list_size;
19     std::list<char> a_list={'a','b','c','d'};
20
21     list_size = a_list.size();
22     for(list<char> :: iterator itr=a_list.begin();itr!=a_list.end();++itr)
23         cout<<*itr;
24
25     cout << "\n";
26
27     list_reverse(a_list);
28
29     for(list<char> :: iterator itr=a_list.begin();itr!=a_list.end();++itr)
30         cout<<*itr;
31
32     cout << "\n";
33 }
```

Run (Ctrl-Enter)

Output Input Comments 0

abcd  
dcba

## Scheme

Beta

paiza.io Scheme Online

Scheme

Main.scm ✕ +

```
1 (define alist '(A B C D) )
2
3 (print alist)
4
5 (define (reverseList l)
6   (define (pop l last)
7     (if (null? l) last
8         (pop (cdr l) (cons (car l) last) ) ) )
9   (pop l ()))
10
11
12
13 (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.