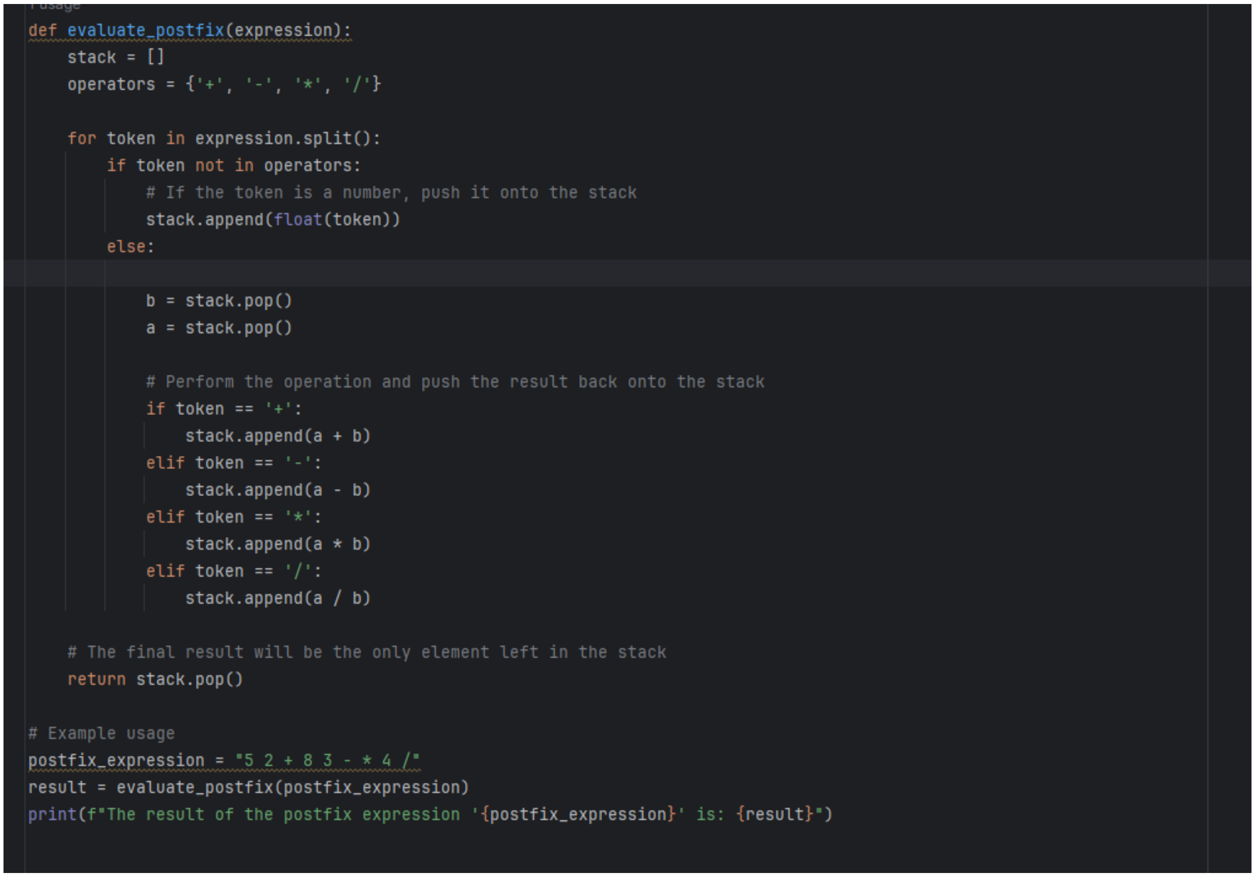
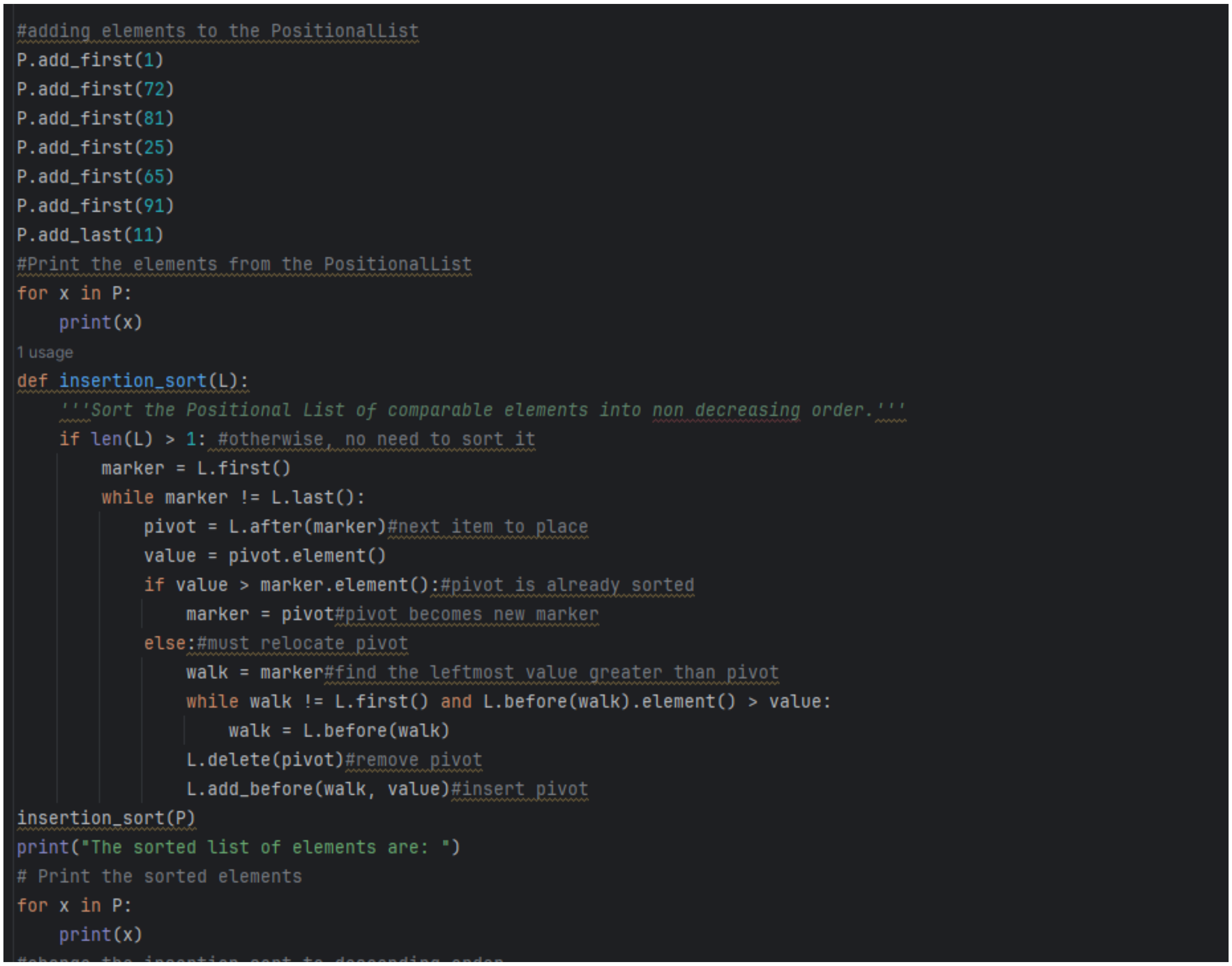
Richmond_Al P. Broqueza 11/22/24

IDB2 DSALGO1

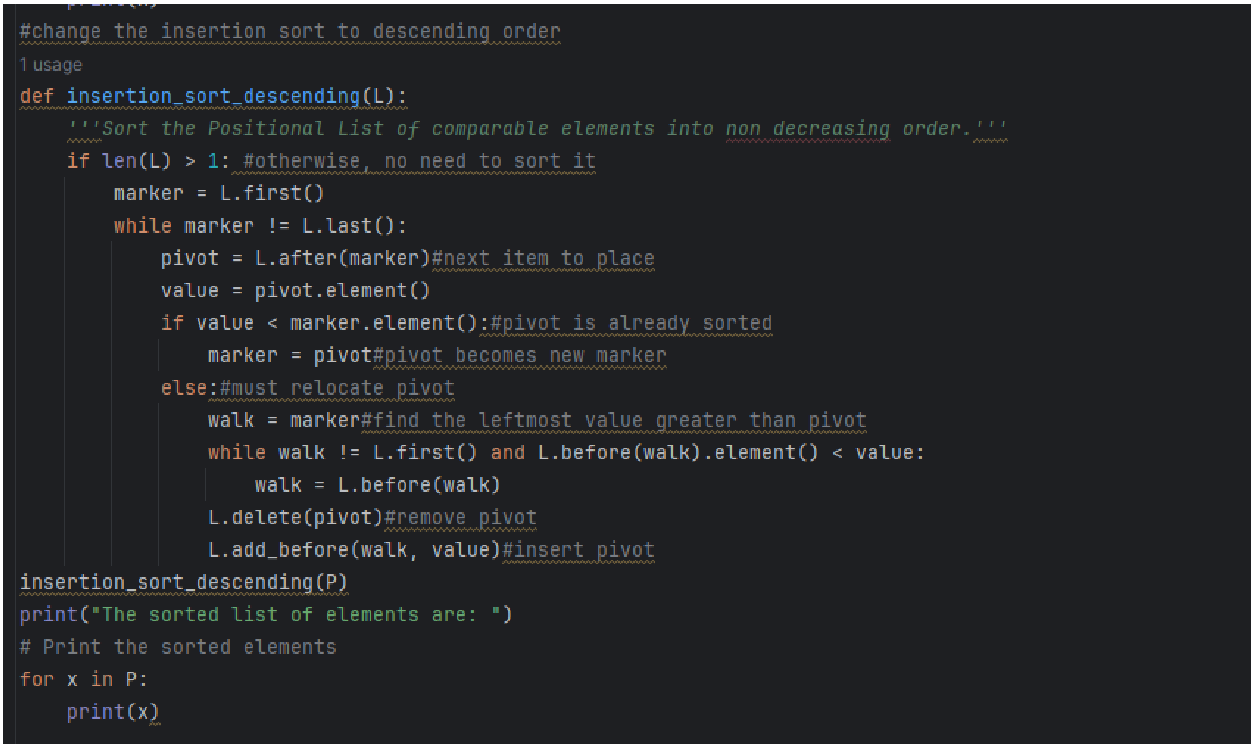
Activity 2 Finals



```
1 usage
def evaluate_postfix(expression):
    stack = []
    operators = {'+', '-', '*', '/'}

    for token in expression.split():
        if token not in operators:
            # If the token is a number, push it onto the stack
            stack.append(float(token))
        else:

            b = stack.pop()
            a = stack.pop()

            # Perform the operation and push the result back onto the stack
            if token == '+':
                stack.append(a + b)
            elif token == '-':
                stack.append(a - b)
            elif token == '*':
                stack.append(a * b)
            elif token == '/':
                stack.append(a / b)

    # The final result will be the only element left in the stack
    return stack.pop()

# Example usage
postfix_expression = "5 2 + 8 3 - * 4 /"
result = evaluate_postfix(postfix_expression)
print(f"The result of the postfix expression '{postfix_expression}' is: {result}")
```

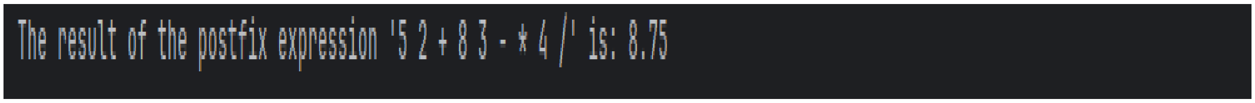


```
#adding elements to the PositionalList
P.add_first(1)
P.add_first(72)
P.add_first(81)
P.add_first(25)
P.add_first(65)
P.add_first(91)
P.add_last(11)
#Print the elements from the PositionalList
for x in P:
    print(x)
1 usage
def insertion_sort(L):
    '''Sort the Positional List of comparable elements into non decreasing order.'''
    if len(L) > 1: #otherwise, no need to sort it
        marker = L.first()
        while marker != L.last():
            pivot = L.after(marker)#next item to place
            value = pivot.element()
            if value > marker.element():#pivot is already sorted
                marker = pivot#pivot becomes new marker
            else:#must relocate pivot
                walk = marker#find the leftmost value greater than pivot
                while walk != L.first() and L.before(walk).element() > value:
                    walk = L.before(walk)
                L.delete(pivot)#remove pivot
                L.add_before(walk, value)#insert pivot
insertion_sort(P)
print("The sorted list of elements are: ")
# Print the sorted elements
for x in P:
    print(x)
#change the insertion sort to descending order
```

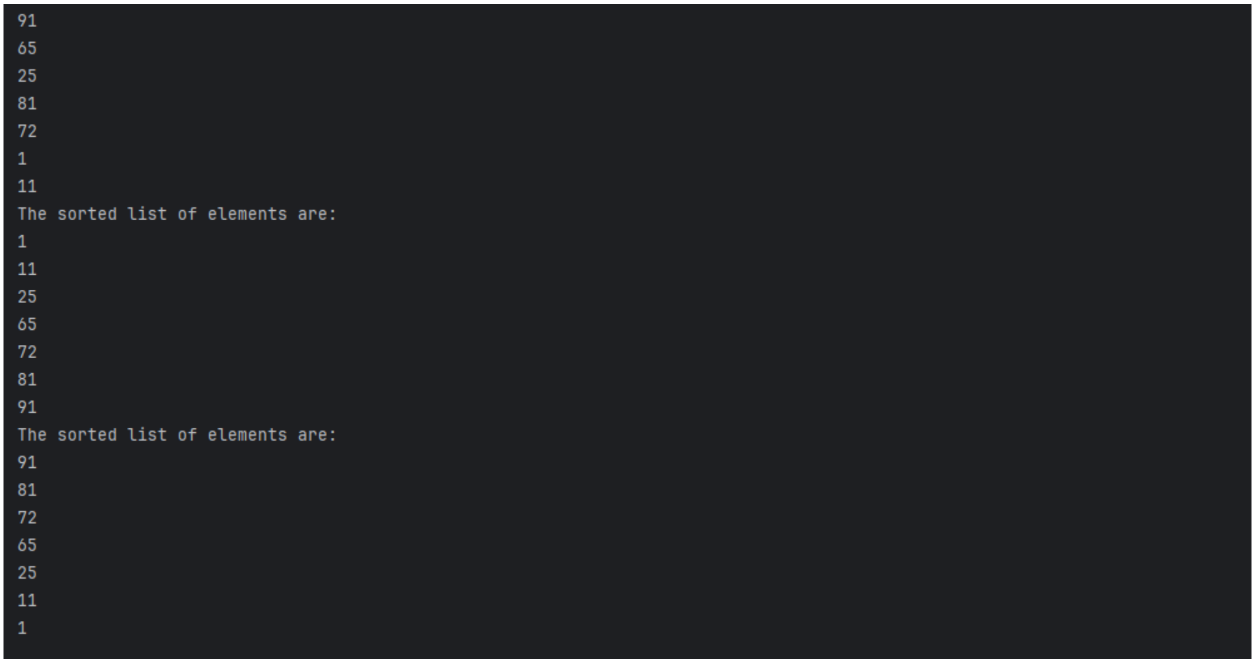


```
#change the insertion sort to descending order
1 usage
def insertion_sort_descending(L):
    '''Sort the Positional List of comparable elements into non decreasing order.'''
    if len(L) > 1: #otherwise, no need to sort it
        marker = L.first()
        while marker != L.last():
            pivot = L.after(marker)#next item to place
            value = pivot.element()
            if value < marker.element():#pivot is already sorted
                marker = pivot#pivot becomes new marker
            else:#must relocate pivot
                walk = marker#find the leftmost value greater than pivot
                while walk != L.first() and L.before(walk).element() < value:
                    walk = L.before(walk)
                L.delete(pivot)#remove pivot
                L.add_before(walk, value)#insert pivot
insertion_sort_descending(P)
print("The sorted list of elements are: ")
# Print the sorted elements
for x in P:
    print(x)
```

Output:



```
The result of the postfix expression '5 2 + 8 3 - * 4 /' is: 8.75
```



```
91
65
25
81
72
1
11
The sorted list of elements are:
1
11
25
65
72
81
91
The sorted list of elements are:
91
81
72
65
25
11
1
```