Galimba, John Rydel	Activity 2 (Finals)
IDB2- DSALGO	11/22/24

```python
2 usages
def prec(c):
    if c == '^':
        return 3
    elif c in ('/', '*'):
        return 2
    elif c in ('+', '-'):
        return 1
    else:
        return -1


1 usage
def infixToPostfix(expression):
    stack = []
    result = ""

    for char in expression:
        if char.isalnum():
            result += char + " "
        elif char == '(':
            stack.append(char)
        elif char == ')':
            while stack and stack[-1] != '(':
                result += stack.pop() + " "
            if stack:
                stack.pop()
        else:
            while stack and prec(char) <= prec(stack[-1]):
                result += stack.pop() + " "
            stack.append(char)

    while stack:
        result += stack.pop() + " "

    return result.strip()

exp = input("Enter an infix expression: ")
postfix_result = infixToPostfix(exp)
print("Postfix expression:", postfix_result)

1 usage
```

```python
def insertion_sort_ascending(arr):
    for i in range(1, len(arr)):
        key = arr[i]
        j = i - 1

        while j >= 0 and key < arr[j]:
            arr[j + 1] = arr[j]
            j -= 1

        arr[j + 1] = key


1 usage
def insertion_sort_descending(arr):
    for i in range(1, len(arr)):
        key = arr[i]
        j = i - 1

        while j >= 0 and key > arr[j]:
            arr[j + 1] = arr[j]
            j -= 1

        arr[j + 1] = key


numbers_input = input("Enter numbers separated by spaces: ")
numbers = list(map(int, numbers_input.split()))

ascending_numbers = numbers.copy()
insertion_sort_ascending(ascending_numbers)
print("Sorted in ascending order:", ascending_numbers)

descending_numbers = numbers.copy()
insertion_sort_descending(descending_numbers)
print("Sorted in descending order:", descending_numbers)
```

OUTPUT

```
"Z:\FinalGrading\Activity #2 (Laboratory)\.venv\Scripts\python.exe" "Z:\Fir
Enter an infix expression: ((5+2)*(8-3))/4
Postfix expression: 5 2 + 8 3 - * 4 /
Enter numbers separated by spaces: 1 72 81 25 65 91 11
Sorted in ascending order: [1, 11, 25, 65, 72, 81, 91]
Sorted in descending order: [91, 81, 72, 65, 25, 11, 1]

Process finished with exit code 0
```