

## Pseudo Code:

## Import math module

Enter Numerator

Enter Denominator

Check if denominator is zero

Output if it is zero

Store absolute values of numerator and denominators

Check if its a proper fraction first by checking absolute value of numerator is smaller than denominator

Calculate modulus of numerator and denominator, then store modulus in a variable

Calculate output denominator by dividing by gcd

## Output the final fractions

## Test Run Screenshots:

Run1:

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows a folder structure under "C:\repos>cis\_homework > New folder > PA1Dangerfield.py".
- Code Editor:** Displays Python code for fraction manipulation. The code handles various cases of fractions, including proper fractions, improper fractions, and mixed numbers. It uses GCD to simplify denominators and handles negative numbers.
- Terminal:** Shows the command used to run the script and the resulting output. The output indicates that 1/2 is a proper fraction.
- Bottom Status Bar:** Shows the file name "git\_test", the current line (Line 42), column (Col 29), and other status information like spaces, tabs, and encoding.

## Run2:

The screenshot shows the VS Code interface with the Python extension installed. The code in the editor is as follows:

```
C:\> repos > cs_homework > pat > PATDangerfield.py
```

```
14
15     #Validate den
16     if den1 == 0:
17         print("denominator cannot be zero.")
18     else:
19         #get absolute value for numbers for later
20         abs_num = abs(num1)
21         abs_den = abs(den1)
22
23     #see if fraction is proper or improper
24     if abs_num < abs_den:
25         #proper fraction
26         print(f"(abs(num1)) / (abs(den1)) is a proper fraction")
27     else:
28         whole = num1 // den1
29         remain = abs(num1 % den1)
30
31     #simplify fraction part if remainder exists
32     if remain != 0:
33         divisor = math.gcd(remain, den1)
34         remain //= divisor
35         simplified_den = den1 // divisor
36
37     else: simplified_den = den1
38
39     #output for mixed or whole number
40     if remain == 0:
41         print(f"(num1) / (den1) is an improper fraction and it can be reduced to {whole}")
42         #output negative improper fractions show as -(a + b/c)
43     elif num1 < 0:
44         print(f"(num1) / (den1) is an improper fraction and its mixed fraction is -((abs(whole)) + (remain) / {simplified_den}).")
45     else:
46         print(f"(num1) / (den1) is an improper fraction and its mixed fraction is {whole} + (remain) / {simplified_den}.")
```

The terminal output shows:

```
PS C:\repos\git_test> & "c:\Users\UncertaintyPrinciple\AppData\Local\Programs\Python\Python314\python.exe" "c:\Users\UncertaintyPrinciple\vscode\extensions\ms-python.python.debug-2025.14.0-win32-x64\bundle\libs\debug\git_launcher" "53280" -- "C:\repos\cs_homework\pat\PATDangerfield.py"
Enter a Numerator: 16
Enter a Denominator: 3
16 / 3 is an improper fraction and its mixed fraction is 5 + 1 / 3.
PS C:\repos\git_test>
```

## Run3:

The screenshot shows the VS Code interface with the Python extension installed. The code in the editor is identical to Run 2:

```
C:\> repos > cs_homework > pat > PATDangerfield.py
```

```
14
15     #Validate den
16     if den1 == 0:
17         print("denominator cannot be zero.")
18     else:
19         #get absolute value for numbers for later
20         abs_num = abs(num1)
21         abs_den = abs(den1)
22
23     #see if fraction is proper or improper
24     if abs_num < abs_den:
25         #proper fraction
26         print(f"(abs(num1)) / (abs(den1)) is a proper fraction")
27     else:
28         whole = num1 // den1
29         remain = abs(num1 % den1)
30
31     #simplify fraction part if remainder exists
32     if remain != 0:
33         divisor = math.gcd(remain, den1)
34         remain //= divisor
35         simplified_den = den1 // divisor
36
37     else: simplified_den = den1
38
39     #output for mixed or whole number
40     if remain == 0:
41         print(f"(num1) / (den1) is an improper fraction and it can be reduced to {whole}")
42         #output negative improper fractions show as -(a + b/c)
43     elif num1 < 0:
44         print(f"(num1) / (den1) is an improper fraction and its mixed fraction is -((abs(whole)) + (remain) / {simplified_den}).")
45     else:
46         print(f"(num1) / (den1) is an improper fraction and its mixed fraction is {whole} + (remain) / {simplified_den}.")
```

The terminal output shows:

```
PS C:\repos\git_test> & "c:\Users\UncertaintyPrinciple\AppData\Local\Programs\Python\Python314\python.exe" "c:\Users\UncertaintyPrinciple\vscode\extensions\ms-python.python.debug-2025.14.0-win32-x64\bundle\libs\debug\git_launcher" "53280" -- "C:\repos\cs_homework\pat\PATDangerfield.py"
Enter a Numerator: 6
Enter a Denominator: 7
6 / 7 is a proper fraction
PS C:\repos\git_test>
```

## Run4:

```
C:\> repos>cs_homework>pat> PAIDangerfield.py >...
14
15     #Validate den1
16     if den1 == 0:
17         print("denominator cannot be zero.")
18     else:
19         #get absolute value for numbers later
20         abs_num = abs(num1)
21         abs_den = abs(den1)
22
23     #see if fraction is proper or improper
24     if abs_num < abs_den:
25         #proper fraction
26         print(f"({abs(num1)}) / ({abs(den1)}) is a proper fraction")
27     else:
28         whole = num1 // den1
29         remain = abs(num1 % den1)
30
31     #simplify fraction part if remainder exists
32     if remain != 0:
33         divisor = math.gcd(remain, den1)
34         remain /= divisor
35         simplified_den = den1 // divisor
36
37     else: simplified_den = den1
38
39     #output for mixed or whole number
40     if remain == 0:
41         print(f"({num1}) / ({den1}) is an improper fraction and it can be reduced to ({whole})")
42         #output negative improper fractions show as -(a + b/c)
43     elif num1 < 0:
44         print(f"({abs(num1)}) / ({den1}) is an improper fraction and its mixed fraction is -(({abs(whole)}) + {remain} / {simplified_den}).")
45     else:
46         print(f"({num1}) / ({den1}) is an improper fraction and its mixed fraction is ({whole}) + {remain} / {simplified_den}.")
47
48
PS C:\repos\git_test> & "c:\Users\UncertaintyPrinciple\AppData\Local\Programs\Python\Python314\python.exe" "c:\Users\UncertaintyPrinciple\.vscode\extensions\ms-python.python.debug-2025.14.0-win32-x64\bundle\dlls\debug\launcher" "5330" "<--" "C:\repos\cs_homework\pat\PAIDangerfield.py"
Enter a Numerator: 6
Enter a Denominator: 2
6 / 2 is an improper fraction and it can be reduced to 3
PS C:\repos\git_test> |
```

## Run 5:

```
C:\> repos>cs_homework>pat> PAIDangerfield.py >...
14
15     #Validate den1
16     if den1 == 0:
17         print("denominator cannot be zero.")
18     else:
19         #get absolute value for numbers later
20         abs_num = abs(num1)
21         abs_den = abs(den1)
22
23     #see if fraction is proper or improper
24     if abs_num < abs_den:
25         #proper fraction
26         print(f"({abs(num1)}) / ({abs(den1)}) is a proper fraction")
27     else:
28         whole = num1 // den1
29         remain = abs(num1 % den1)
30
31     #simplify fraction part if remainder exists
32     if remain != 0:
33         divisor = math.gcd(remain, den1)
34         remain /= divisor
35         simplified_den = den1 // divisor
36
37     else: simplified_den = den1
38
39     #output for mixed or whole number
40     if remain == 0:
41         print(f"({num1}) / ({den1}) is an improper fraction and it can be reduced to ({whole})")
42         #output negative improper fractions show as -(a + b/c)
43     elif num1 < 0:
44         print(f"({abs(num1)}) / ({den1}) is an improper fraction and its mixed fraction is -(({abs(whole)}) + {remain} / {simplified_den}).")
45     else:
46         print(f"({num1}) / ({den1}) is an improper fraction and its mixed fraction is ({whole}) + {remain} / {simplified_den}.")
47
48
PS C:\repos\git_test> & "c:\Users\UncertaintyPrinciple\AppData\Local\Programs\Python\Python314\python.exe" "c:\Users\UncertaintyPrinciple\.vscode\extensions\ms-python.python.debug-2025.14.0-win32-x64\bundle\dlls\debug\launcher" "5330" "<--" "C:\repos\cs_homework\pat\PAIDangerfield.py"
Enter a Numerator: 6
Enter a Denominator: 2
6 / 2 is a proper fraction
PS C:\repos\git_test> |
```