

## euclidclient1.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # euclidclient1.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9:
10: #-----
11:
12: def gcd(i, j):
13:
14:     i = abs(i)
15:     j = abs(j)
16:     while j != 0: # Euclid's algorithm
17:         i, j = j, i%j
18:     return i
19:
20: #-----
21:
22: def lcm(i, j):
23:
24:     i = abs(i)
25:     j = abs(j)
26:     return (i // gcd(i, j)) * j
27:
28: #-----
29:
30: def main():
31:
32:     try:
33:         line = input('Enter the first integer: ')
34:         i = int(line)
35:
36:         line = input('Enter the second integer: ')
37:         j = int(line)
38:
39:         my_gcd = gcd(i, j)
40:         print('gcd:', my_gcd)
41:
42:         my_lcm = lcm(i, j)
43:         print('lcm:', my_lcm)
44:
45:     except ValueError:
46:         print('Error: Not an integer', file=sys.stderr)
47:         sys.exit(1)
48:     except EOFError:
49:         print('Error: Missing integer', file=sys.stderr)
50:         sys.exit(1)
51:
52: if __name__ == '__main__':
53:     main()

```

## euclidclient2.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # euclidclient2.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9:
10: #-----
11:
12: def gcd(i, j):
13:
14:     if (i == 0) and (j == 0):
15:         raise ZeroDivisionError(
16:             'gcd(i,j) is undefined if i and j are 0')
17:     i = abs(i)
18:     j = abs(j)
19:     while j != 0: # Euclid's algorithm
20:         i, j = j, i%j
21:     return i
22:
23: #-----
24:
25: def lcm(i, j):
26:
27:     if (i == 0) or (j == 0):
28:         raise ZeroDivisionError(
29:             'lcm(i,j) is undefined if i or j is 0')
30:     i = abs(i)
31:     j = abs(j)
32:     return (i // gcd(i, j)) * j
33:
34: #-----
35:
36: def main():
37:
38:     try:
39:         line = input('Enter the first integer: ')
40:         i = int(line)
41:
42:         line = input('Enter the second integer: ')
43:         j = int(line)
44:
45:         my_gcd = gcd(i, j)
46:         print('gcd:', my_gcd)
47:
48:         my_lcm = lcm(i, j)
49:         print('lcm:', my_lcm)
50:
51:     except ValueError:
52:         print('Error: Not an integer', file=sys.stderr)
53:         sys.exit(1)
54:     except EOFError:
55:         print('Error: Missing integer', file=sys.stderr)
56:         sys.exit(1)
57:     except ZeroDivisionError as ex:
58:         print(str(ex), file=sys.stderr)
59:         sys.exit(1)
60:
61: if __name__ == '__main__':
62:     main()

```

## euclid.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # euclid.py
5: # Author: Bob Dondero
6: #-----
7:
8: def gcd(i, j):
9:
10:     if (i == 0) and (j == 0):
11:         raise ZeroDivisionError(
12:             'gcd(i,j) is undefined if i and j are 0')
13:     i = abs(i)
14:     j = abs(j)
15:     while j != 0: # Euclid's algorithm
16:         i, j = j, i%j
17:     return i
18:
19: #-----
20:
21: def lcm(i, j):
22:
23:     if (i == 0) or (j == 0):
24:         raise ZeroDivisionError(
25:             'lcm(i,j) is undefined if i or j is 0')
26:     i = abs(i)
27:     j = abs(j)
28:     return (i // gcd(i, j)) * j

```

## euclidclient3.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # euclidclient3.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import euclid
10:
11: def main():
12:
13:     try:
14:         line = input('Enter the first integer: ')
15:         i = int(line)
16:
17:         line = input('Enter the second integer: ')
18:         j = int(line)
19:
20:         my_gcd = euclid.gcd(i, j)
21:         print('gcd:', my_gcd)
22:
23:         my_lcm = euclid.lcm(i, j)
24:         print('lcm:', my_lcm)
25:
26:     except ValueError:
27:         print('Error: Not an integer', file=sys.stderr)
28:         sys.exit(1)
29:     except EOFError:
30:         print('Error: Missing integer', file=sys.stderr)
31:         sys.exit(1)
32:     except ZeroDivisionError as ex:
33:         print(str(ex), file=sys.stderr)
34:         sys.exit(1)
35:
36: if __name__ == '__main__':
37:     main()

```

## intmath/\_\_init\_\_.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:

```

## intmath/euclid.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # euclid.py
5: # Author: Bob Dondero
6: #-----
7:
8: def gcd(i, j):
9:
10:     if (i == 0) and (j == 0):
11:         raise ZeroDivisionError(
12:             'gcd(i,j) is undefined if i and j are 0')
13:
14:     i = abs(i)
15:     j = abs(j)
16:     while j != 0: # Euclid's algorithm
17:         i, j = j, i%j
18:     return i
19:
20: #-----
21:
22: def lcm(i, j):
23:
24:     if (i == 0) or (j == 0):
25:         raise ZeroDivisionError(
26:             'lcm(i,j) is undefined if i or j is 0')
27:
28:     i = abs(i)
29:     j = abs(j)
30:     return (i // gcd(i, j)) * j

```

**euclidclient4.py (Page 1 of 1)**

```

1: #!/usr/bin/env python
2:
3: #-----
4: # euclidclient4.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import intmath.euclid
10:
11: def main():
12:
13:     try:
14:         line = input('Enter the first integer: ')
15:         i = int(line)
16:
17:         line = input('Enter the second integer: ')
18:         j = int(line)
19:
20:         my_gcd = intmath.euclid.gcd(i, j)
21:         print('gcd:', my_gcd)
22:
23:         my_lcm = intmath.euclid.lcm(i, j)
24:         print('lcm:', my_lcm)
25:
26:     except ValueError:
27:         print('Error: Not an integer', file=sys.stderr)
28:         sys.exit(1)
29:     except EOFError:
30:         print('Error: Missing integer', file=sys.stderr)
31:         sys.exit(1)
32:     except ZeroDivisionError as ex:
33:         print(str(ex), file=sys.stderr)
34:         sys.exit(1)
35:
36: if __name__ == '__main__':
37:     main()

```

**blank (Page 1 of 1)**

```

1: This page is intentionally blank.

```

## fractionprelim.py (Page 1 of 2)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # fractionprelim.py
5: # Author: Bob Dondero
6: #-----
7:
8: import euclid
9:
10: #-----
11:
12: class Fraction:
13:
14:     def __init__(self, num=0, den=1):
15:         if den == 0:
16:             raise ZeroDivisionError('Denominator cannot be 0')
17:         self._num = num
18:         self._den = den
19:         self._normalize()
20:
21:     def _normalize(self):
22:         if self._den < 0:
23:             self._num *= -1
24:             self._den *= -1
25:         if self._num == 0:
26:             self._den = 1
27:         else:
28:             gcden = euclid.gcd(self._num, self._den)
29:             self._num //= gcden
30:             self._den //= gcden
31:
32:     def to_string(self):
33:         if self._den == 1:
34:             return str(self._num)
35:         return '%d/%d' % (self._num, self._den)
36:
37:     def equals(self, other):
38:         return (self._num == other._num) and (self._den == other._den)
39:
40:     def compare_to(self, other):
41:         if (self._num * other._den) < (other._num * self._den):
42:             return -1
43:         if (self._num * other._den) > (other._num * self._den):
44:             return 1
45:         return 0
46:
47:     def negate(self):
48:         return Fraction(-self._num, self._den)
49:
50:     def add(self, other):
51:         new_num = (self._num * other._den) + (other._num * self._den)
52:         new_den = self._den * other._den
53:         return Fraction(new_num, new_den)
54:
55:     def subtract(self, other):
56:         new_num = (self._num * other._den) - (other._num * self._den)
57:         new_den = self._den * other._den
58:         return Fraction(new_num, new_den)
59:
60:     def multiply(self, other):
61:         new_num = self._num * other._num
62:         new_den = self._den * other._den
63:         return Fraction(new_num, new_den)
64:
65:     def divide(self, other):

```

## fractionprelim.py (Page 2 of 2)

```

66:         if other.equals(Fraction(0, 1)):
67:             raise ZeroDivisionError('Divisor cannot be 0')
68:         new_num = self._num * other._den
69:         new_den = self._den * other._num
70:         return Fraction(new_num, new_den)

```

## fractionprelimclient.py (Page 1 of 2)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # fraction1client.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import fractionprelim as fraction
10:
11: #-----
12:
13: def main():
14:
15:     try:
16:
17:         line = input('Numerator 1: ')
18:         num1 = int(line)
19:         line = input('Denominator 1: ')
20:         den1 = int(line)
21:         line = input('Numerator 2: ')
22:         num2 = int(line)
23:         line = input('Denominator 2: ')
24:         den2 = int(line)
25:
26:         frac1 = fraction.Fraction(num1, den1)
27:         print('frac1:', frac1.to_string())
28:
29:         frac2 = fraction.Fraction(num2, den2)
30:         print('frac2:', frac2.to_string())
31:
32:         if frac1.equals(frac2):
33:             print('frac1 equals frac2')
34:         if not frac1.equals(frac2):
35:             print('frac1 does not equal frac2')
36:
37:         comparison = frac1.compare_to(frac2)
38:         if comparison < 0:
39:             print('frac1 is less than frac2')
40:         if comparison > 0:
41:             print('frac1 is greater than frac2')
42:         if comparison <= 0:
43:             print('frac1 is less than or equal to frac2')
44:         if comparison >= 0:
45:             print('frac1 is greater than or equal to frac2')
46:
47:         frac3 = frac1.negate()
48:         print('-frac1:', frac3.to_string())
49:
50:         frac3 = frac1.add(frac2)
51:         print('frac1 + frac2:', frac3.to_string())
52:
53:         frac3 = frac1.subtract(frac2)
54:         print('frac1 - frac2:', frac3.to_string())
55:
56:         frac3 = frac1.multiply(frac2)
57:         print('frac1 * frac2:', frac3.to_string())
58:
59:         frac3 = frac1.divide(frac2)
60:         print('frac1 / frac2:', frac3.to_string())
61:
62:     except ValueError:
63:         print('Error: Not an integer', file=sys.stderr)
64:         sys.exit(1)
65:     except EOFError:

```

## fractionprelimclient.py (Page 2 of 2)

```

66:         print('Error: Missing integer', file=sys.stderr)
67:         sys.exit(1)
68:     except Exception as ex:
69:         print(str(ex), file=sys.stderr)
70:         sys.exit(1)
71:
72: #-----
73:
74: if __name__ == '__main__':
75:     main()

```

## euclidstrong.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # euclidstrong.py
5: # Author: Bob Dondero
6: #-----
7:
8: def gcd(i, j):
9:
10:     if not isinstance(i, int) or not isinstance(j, int):
11:         raise TypeError('gcd() arguments must be integers')
12:
13:     if (i == 0) and (j == 0):
14:         raise ZeroDivisionError(
15:             'gcd(i,j) is undefined if i and j are 0')
16:
17:     i = abs(i)
18:     j = abs(j)
19:     while j != 0: # Euclid's algorithm
20:         i, j = j, i%j
21:     return i
22:
23: #-----
24:
25: def lcm(i, j):
26:
27:     if not isinstance(i, int) or not isinstance(j, int):
28:         raise TypeError('lcm() arguments must be integers')
29:
30:     if (i == 0) or (j == 0):
31:         raise ZeroDivisionError(
32:             'lcm(i,j) is undefined if i or j is 0')
33:
34:     i = abs(i)
35:     j = abs(j)
36:     return (i // gcd(i, j)) * j

```