intmath1/testintmath.c (Page 1 of 2)

```
 1: /*--------------------------------------------------------------------*/
 2: /* testintmath.c (Version 1)                                          */
 3: /* Author: Bob Dondero                                                */
 4: /*--------------------------------------------------------------------*/
 5:
 6: #include <stdio.h>
 7: #include <stdlib.h>
 8:
 9: /*--------------------------------------------------------------------*/
10:
11: /* Return the greatest common divisor of positive integers iFirst and
12:    iSecond. */
13:
14: static int gcd(int iFirst, int iSecond)
15: {
16:    int iTemp;
17:
18:    /* Use Euclid's algorithm. */
19:
20:    while (iSecond != 0)
21:    {
22:       iTemp = iFirst % iSecond;
23:       iFirst = iSecond;
24:       iSecond = iTemp;
25:    }
26:
27:    return iFirst;
28: }
29:
30: /*--------------------------------------------------------------------*/
31:
32: /* Return the least common multiple of positive integers iFirst and
33:    iSecond. */
34:
35: static int lcm(int iFirst, int iSecond)
36: {
37:    return (iFirst / gcd(iFirst, iSecond)) * iSecond;
38: }
39:
40: /*--------------------------------------------------------------------*/
41:
42: /* Read two positive integers from stdin. Return EXIT_FAILURE if stdin
43:    contains bad data. Otherwise compute the greatest common divisor
44:    and least common multiple of the two positive integers, write those
45:    two values to stdout, and return 0. */
46:
47: int main(void)
48: {
49:    int i1;
50:    int i2;
51:    int iGcd;
52:    int iLcm;
53:    int iScanfReturn;
54:
55:    printf("Enter the first positive integer:\n");
56:    iScanfReturn = scanf("%d", &i1);
57:    if ((iScanfReturn != 1) || (i1 <= 0))
58:    {
59:       fprintf(stderr, "Error: Not a positive integer.\n");
60:       exit(EXIT_FAILURE);
61:    }
62:
63:    printf("Enter the second positive integer:\n");
```

intmath1/testintmath.c (Page 2 of 2)

```
64:     iScanfReturn = scanf("%d", &i2);
65:     if ((iScanfReturn != 1) || (i2 <= 0))
66:     {
67:        fprintf(stderr, "Error: Not a positive integer.\n");
68:        exit(EXIT_FAILURE);
69:     }
70:
71:     iGcd = gcd(i1, i2);
72:     iLcm = lcm(i1, i2);
73:
74:     printf("The greatest common divisor of %d and %d is %d.\n",
75:        i1, i2, iGcd);
76:     printf("The least common multiple of %d and %d is %d.\n",
77:        i1, i2, iLcm);
78:
79:     return 0;
80: }
```

intmath2/testintmath.c (Page 1 of 2)

```
 1: /*--------------------------------------------------------------------*/
 2: /* testintmath.c (Version 2)                                          */
 3: /* Author: Bob Dondero                                                */
 4: /*--------------------------------------------------------------------*/
 5:
 6: #include <stdio.h>
 7: #include <stdlib.h>
 8:
 9: /*--------------------------------------------------------------------*/
10: /* Function declarations                                              */
11: /*--------------------------------------------------------------------*/
12:
13: /* Return the greatest common divisor of positive integers iFirst and
14:    iSecond. */
15: static int gcd(int iFirst, int iSecond);
16:
17: /* Return the least common multiple of positive integers iFirst and
18:    iSecond. */
19: static int lcm(int iFirst, int iSecond);
20:
21: /*--------------------------------------------------------------------*/
22: /* Function definitions                                               */
23: /*--------------------------------------------------------------------*/
24:
25: /* Read two positive integers from stdin. Return EXIT_FAILURE if stdin
26:    contains bad data. Otherwise compute the greatest common divisor
27:    and least common multiple of the two positive integers, write those
28:    two values to stdout, and return 0. */
29:
30: int main(void)
31: {
32:    int i1;
33:    int i2;
34:    int iGcd;
35:    int iLcm;
36:    int iScanfReturn;
37:
38:    printf("Enter the first positive integer:\n");
39:    iScanfReturn = scanf("%d", &i1);
40:    if ((iScanfReturn != 1) || (i1 <= 0))
41:    {
42:       fprintf(stderr, "Error: Not a positive integer.\n");
43:       exit(EXIT_FAILURE);
44:    }
45:
46:    printf("Enter the second positive integer:\n");
47:    iScanfReturn = scanf("%d", &i2);
48:    if ((iScanfReturn != 1) || (i2 <= 0))
49:    {
50:       fprintf(stderr, "Error: Not a positive integer.\n");
51:       exit(EXIT_FAILURE);
52:    }
53:
54:    iGcd = gcd(i1, i2);
55:    iLcm = lcm(i1, i2);
56:
57:    printf("The greatest common divisor of %d and %d is %d.\n",
58:       i1, i2, iGcd);
59:    printf("The least common multiple of %d and %d is %d.\n",
60:       i1, i2, iLcm);
61:
62:    return 0;
63: }
```

intmath2/testintmath.c (Page 2 of 2)

```
64:
65: /*--------------------------------------------------------------------*/
66:
67: static int gcd(int iFirst, int iSecond)
68: {
69:    int iTemp;
70:
71:    /* Use Euclid's algorithm. */
72:
73:    while (iSecond != 0)
74:    {
75:       iTemp = iFirst % iSecond;
76:       iFirst = iSecond;
77:       iSecond = iTemp;
78:    }
79:
80:    return iFirst;
81: }
82:
83: /*--------------------------------------------------------------------*/
84:
85: static int lcm(int iFirst, int iSecond)
86: {
87:    return (iFirst / gcd(iFirst, iSecond)) * iSecond;
88: }
```

intmath3/intmath.h (Page 1 of 1)

```
 1: /*-------------------------------------------------------------------------*/
 2: /* intmath.h (Version 3)                                                   */
 3: /* Author: Bob Dondero                                                     */
 4: /*-------------------------------------------------------------------------*/
 5:
 6: /* Return the greatest common divisor of positive integers iFirst and
 7:    iSecond. */
 8:
 9: int gcd(int iFirst, int iSecond);
10:
11: /*-------------------------------------------------------------------------*/
12:
13: /* Return the least common multiple of positive integers iFirst and
14:    iSecond. */
15:
16: int lcm(int iFirst, int iSecond);
```

intmath3/intmath.c (Page 1 of 1)

```
 1: /*-------------------------------------------------------------------------*/
 2: /* intmath.c (Version 3)                                                   */
 3: /* Author: Bob Dondero                                                     */
 4: /*-------------------------------------------------------------------------*/
 5:
 6: #include "intmath.h"
 7:
 8: /*-------------------------------------------------------------------------*/
 9:
10: int gcd(int iFirst, int iSecond)
11: {
12:    int iTemp;
13:
14:    /* Use Euclid's algorithm. */
15:
16:    while (iSecond != 0)
17:    {
18:       iTemp = iFirst % iSecond;
19:       iFirst = iSecond;
20:       iSecond = iTemp;
21:    }
22:
23:    return iFirst;
24: }
25:
26: /*-------------------------------------------------------------------------*/
27:
28: int lcm(int iFirst, int iSecond)
29: {
30:    return (iFirst / gcd(iFirst, iSecond)) * iSecond;
31: }
```

intmath3/testintmath.c (Page 1 of 1)

```
 1: /*--------------------------------------------------------------------*/
 2: /* testintmath.c (Version 3)                                          */
 3: /* Author: Bob Dondero                                                */
 4: /*--------------------------------------------------------------------*/
 5:
 6: #include "intmath.h"
 7: #include <stdio.h>
 8: #include <stdlib.h>
 9:
10: /*--------------------------------------------------------------------*/
11:
12: /* Read two positive integers from stdin. Return EXIT_FAILURE if stdin
13:    contains bad data. Otherwise compute the greatest common divisor
14:    and least common multiple of the two positive integers, write those
15:    two values to stdout, and return 0. */
16:
17: int main(void)
18: {
19:    int i1;
20:    int i2;
21:    int iGcd;
22:    int iLcm;
23:    int iScanfReturn;
24:
25:    printf("Enter the first positive integer:\n");
26:    iScanfReturn = scanf("%d", &i1);
27:    if ((iScanfReturn != 1) || (i1 <= 0))
28:    {
29:       fprintf(stderr, "Error: Not a positive integer.\n");
30:       exit(EXIT_FAILURE);
31:    }
32:
33:    printf("Enter the second positive integer:\n");
34:    iScanfReturn = scanf("%d", &i2);
35:    if ((iScanfReturn != 1) || (i2 <= 0))
36:    {
37:       fprintf(stderr, "Error: Not a positive integer.\n");
38:       exit(EXIT_FAILURE);
39:    }
40:
41:    iGcd = gcd(i1, i2);
42:    iLcm = lcm(i1, i2);
43:
44:    printf("The greatest common divisor of %d and %d is %d.\n",
45:       i1, i2, iGcd);
46:    printf("The least common multiple of %d and %d is %d.\n",
47:       i1, i2, iLcm);
48:
49:    return 0;
50: }
```

intmath4/intmath.h (Page 1 of 1)

```
 1: /*-------------------------------------------------------------------------*/
 2: /* intmath.h (Version 4)                                                   */
 3: /* Author: Bob Dondero                                                     */
 4: /*-------------------------------------------------------------------------*/
 5:
 6: #ifndef INTMATH_INCLUDED
 7: #define INTMATH_INCLUDED
 8:
 9: /*-------------------------------------------------------------------------*/
10:
11: /* Return the greatest common divisor of positive integers iFirst and
12:    iSecond. */
13:
14: int IntMath_gcd(int iFirst, int iSecond);
15:
16: /*-------------------------------------------------------------------------*/
17:
18: /* Return the least common multiple of positive integers iFirst and
19:    iSecond. */
20:
21: int IntMath_lcm(int iFirst, int iSecond);
22:
23: #endif
```

intmath4/intmath.c (Page 1 of 1)

```
 1: /*--------------------------------------------------------------------*/
 2: /* intmath.c (Version 4)                                              */
 3: /* Author: Bob Dondero                                                */
 4: /*--------------------------------------------------------------------*/
 5:
 6: #include "intmath.h"
 7: #include <assert.h>
 8:
 9: /*--------------------------------------------------------------------*/
10:
11: int IntMath_gcd(int iFirst, int iSecond)
12: {
13:    int iTemp;
14:
15:    assert(iFirst > 0);
16:    assert(iSecond > 0);
17:
18:    /* Use Euclid's algorithm. */
19:
20:    while (iSecond != 0)
21:    {
22:      iTemp = iFirst % iSecond;
23:      iFirst = iSecond;
24:      iSecond = iTemp;
25:    }
26:
27:    return iFirst;
28: }
29:
30: /*--------------------------------------------------------------------*/
31:
32: int IntMath_lcm(int iFirst, int iSecond)
33: {
34:    assert(iFirst > 0);
35:    assert(iSecond > 0);
36:
37:    return (iFirst / IntMath_gcd(iFirst, iSecond)) * iSecond;
38: }
```

intmath4/testintmath.c (Page 1 of 1)

```
 1: /*--------------------------------------------------------------------*/
 2: /* testintmath.c (Version 4)                                          */
 3: /* Author: Bob Dondero                                                */
 4: /*--------------------------------------------------------------------*/
 5:
 6: #include "intmath.h"
 7: #include <stdio.h>
 8: #include <stdlib.h>
 9:
10: /*--------------------------------------------------------------------*/
11:
12: /* Read two positive integers from stdin. Return EXIT_FAILURE if stdin
13:    contains bad data. Otherwise compute the greatest common divisor
14:    and least common multiple of the two positive integers, write those
15:    two values to stdout, and return 0. */
16:
17: int main(void)
18: {
19:    int i1;
20:    int i2;
21:    int iGcd;
22:    int iLcm;
23:    int iScanfReturn;
24:
25:    printf("Enter the first positive integer:\n");
26:    iScanfReturn = scanf("%d", &i1);
27:    if ((iScanfReturn != 1) || (i1 <= 0))
28:    {
29:       fprintf(stderr, "Error: Not a positive integer.\n");
30:       exit(EXIT_FAILURE);
31:    }
32:
33:    printf("Enter the second positive integer:\n");
34:    iScanfReturn = scanf("%d", &i2);
35:    if ((iScanfReturn != 1) || (i2 <= 0))
36:    {
37:       fprintf(stderr, "Error: Not a positive integer.\n");
38:       exit(EXIT_FAILURE);
39:    }
40:
41:    iGcd = IntMath_gcd(i1, i2);
42:    iLcm = IntMath_lcm(i1, i2);
43:
44:    printf("The greatest common divisor of %d and %d is %d.\n",
45:       i1, i2, iGcd);
46:    printf("The least common multiple of %d and %d is %d.\n",
47:       i1, i2, iLcm);
48:
49:    return 0;
50: }
```