Practice problems on loops and nested loops: Write C++ programs to produce the output described in each of the following problems.

The appropriate loop of a C++ program is shown for each example. In each case, this loop should be placed into a C++ program along with any necessary declarations of variables.

For each problem there are many other ways to write the loops.

(1) A triangle of stars (the user supplies the height):

```
*
**
***
****
```

```
for (r = 1; r <= max; r++) {
  for (c = 1; c <= r; c++)
     cout << "*";
  cout << endl;
}
```

(2) Another triangle of stars (the user supplies the height)

```
      *
     ***
    *****
   *******
```

```
spaces = max;
for (r = 1; r <= max; r++) {
  spaces--;
  for (c = 1; c <= spaces; c++)
     cout << " ";
  for (c = 1; c <= 2*r - 1; c++)
     cout << "*";
  cout << endl;
}
```

(3) Another triangle of stars (the user supplies the height)

```
       *
      **
     ***
    ****
     ***
      **
       *
```

```
spaces = max;
rows = max + max - 1;
for (r = 1; r <= rows; r++) {
  if (r <= max) spaces--;
  else spaces++;
  for (c = 1; c <= spaces; c++)
     cout << " ";
  for (c = spaces + 1; c <= max; c++)
     cout << "*";
  cout << endl;
}
```

(4) A double rectangle of stars (the user supplies height and width)

```
  * * * * * *
   * * * * *
  * * * * * *
   * * * * *
  * * * * * *
```

```
for (r = 1; r <= rows; r++) {
  for (c = 1; c <= cols; c++)
     cout << "* ";
  cout << endl;
  if (r < rows) {
     for (c = 1; c <= cols -1; c++)
        cout << " *";
     cout << endl;
  }
}
```

(5) A table of powers of 2 (user supplies the number)

```
   2 to the power 0 is 1
   2 to the power 1 is 2
   2 to the power 2 is 4
   2 to the power 3 is 8
```

```
power = 1;
for (r = 0; r <= max; r++) {
  cout << "2 to the power " << r << " is " << power << endl;
  power *= 2;
}
```

(6) A table of factorials (user supplies the number)

```
   1! = 1 = 1
   2! = 2 x 1 = 2
   3! = 3 x 2 x 1 = 6
   4! = 4 x 3 x 2 x 1 = 24
   5! = 5 x 4 x 3 x 2 x 1 = 120
```

```
factorial = 1;
for (r = 1; r <= max; r++) {
  cout << r << "! = ";
  for (c = r; c >= 1; c--) {
    cout << c;
    if (c > 1) cout << " x ";
    else cout << " = ";
  }
  factorial *= r;
  cout << factorial << endl;
}
```

Practice problems on for loops, while loops, and nested loops: Write C++ programs to produce the output described in each of the following problems.

(1) The largest factor of a number (the user supplies the number):

number = 24, largest factor = 12

Answer:

```
int main() {
   int number;
   cout << "Enter an integer that is larger than 1: ";
   cin >> number;
   if (number <= 1) exit(1);

   int factor = number -1;
   while ((number % factor) > 0) factor--;
   cout << "The largest factor is: " << factor << endl;
   return 0;
}
```

(2) An empty diamond of stars (the user supplies the height)

```
      *
     * *
    *   *
   *     *
    *   *
     * *
      *
```

Answer:

```
int main() {
   int r, c, ht;
   cout << "Enter an odd integer as the height: ";
   cin >> ht;
   if ((ht<= 0) || ((ht % 2) == 0)) exit(1);
   int mid = ht / 2 + 1;

   for (r = mid; r >= 1; r--) {
      for (c = 1; c <= ht; c++) {
         if (c == r || c == (ht + 1 - r)) cout << "*";
         else cout << " ";
      }
      cout << endl;
   }

   for (r = 2; r <= mid; r++) {
      for (c = 1; c <= ht; c++) {
         if (c == r || c == (ht + 1 - r)) cout << "*";
         else cout << " ";
      }
      cout << endl;
   }

   return 0;
}
```

(3) An X of stars (the user supplies the height)

```
*     *
 *   *
  * *
   *
  * *
 *   *
*     *
```

```
int main() {
   int r, c, ht;
   cout << "Enter an odd integer as the height: ";
   cin >> ht;
   if ((ht<= 0) || ((ht % 2) == 0)) exit(1);

   for (r = 1; r <= ht; r++) {
      for (c = 1; c <= ht; c++) {
         if (c == r || c == (ht + 1 - r)) cout << "*";
         else cout << " ";
      }
      cout << endl;
   }

   return 0;
}
```

(4) A pattern of 0's surrounded by *'s (the user supplies height and width)

```
***********
*0*0*0*0*0*
***********
*0*0*0*0*0*
***********
```

```
int main() {
   int r, c, ht, width;
   cout << "Enter a height and width: ";
   cin >> ht >> width;

   for (r = 1; r <= ht; r++) {
      for (c = 1; c <= width; c++) {
         if ((r % 2 == 0) && (c % 2 == 0))
           cout << "0";
         else cout << "*";
      }
      cout << endl;
   }

   return 0;
}
```

(5) A multiplication table (user supplies the size)

```
 1  2  3  4  5  6
```

```
    -----------------
1*  1  2  3  4  5  6
2*  2  4  6  8 10 12
3*  3  6  9 12 15 18
4*  4  8 12 16 20 24
5*  5 10 15 20 25 30
6*  6 12 18 24 30 36
```

```
int main() {
   int r, c, size;
   cout << "Enter a positive integer as the size: ";
   cin >> size;
   if (size <= 0) exit(1);

// output here is designed for sizes up to 10
   cout << "    ";
   for (c = 1; c <= size; c++)
      cout << " " << c << " ";
   cout << endl;

   cout << "    ";
   for (c = 1; c <= size; c++)
      cout << "---";
   cout << endl;

   for (r = 1; r <= size; r++) {
      cout << r << "*  ";
      for (c = 1; c <= size; c++) {
         if (r * c < 10) cout << " ";
         cout << r * c  << " ";
      }
      cout << endl;
   }

   return 0;
}
```

(6) Sum of the digits of a number (user supplies the number)

```
number = 245,  sum of digits = 11
```

```
int main() {
   int n;
   cout << "Enter a positive integer: ";
   cin >> n;
   if (n <= 0) exit(1);

   int sum = 0;
   while (n > 0) {
      sum = sum + n % 10;
      n = n / 10;
   }
   cout << "The digit sum is: " << sum << endl;

   return 0;
}
```