**Exercise 1：Please write programs to output following special figures.**

**Task 1:**

Input an integer *n*, output *n* characters ‘\*’ in one line.

For example, when *n* is 3, The output is \*\*\*. When *n* is 4, The Output is \*\*\*\*.

**Task 2:**

Input an integer *n*, output *n\*n* matrix made of characters ‘\*’.

For example, when *n* is 3, The output is

\*\*\*

\*\*\*

\*\*\*

When *n* is 4, The Output is

\*\*\*\*

\*\*\*\*

\*\*\*\*

\*\*\*\*

**Task 3:**

Input an integer *n*, output a triangle made of characters ‘\*’.

For example, when *n* is 3, The output is

\*

\*\*

\*\*\*

When *n* is 4, The Output is

\*

\*\*

\*\*\*

\*\*\*\*

(If you have any questions as you proceed, please read the appendix first)

**Task 4:**

Input an integer *n*, output a triangle made of characters ‘\*’.

For example, when *n* is 3, The output is

\*

\*\*\*

\*\*\*\*\*

When *n* is 4, The Output is

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

**Exercise 2：Please write programs to output following special figures.**

**Task 1:**

Input an integer *n*, output 1~*n* in one line.

For example, when *n* is 3, The output is 1 2 3. When *n* is 4, The Output is 1 2 3 4.

**Task 2:**

Input an integer *n*, output *n\*n* matrix made of numbers 1~*n*.

For example, when *n* is 3, The output is

1 2 3

1 2 3

1 2 3

When *n* is 4, The Output is

1 2 3 4

1 2 3 4

1 2 3 4

1 2 3 4

**Task 3:**

Input an integer *n*, output *n\*n* matrix made of numbers 1~*n\*n*.

For example, when *n* is 3, The output is

1 2 3

4 5 6

7 8 9

When *n* is 4, The Output is

1 2 3 4

5 6 7 8

9 10 11 12

13 14 15 16

**Task 4:**

Input an integer *n*, output a triangle made of numbers.

For example, when *n* is 3, The output is

1

2 3

4 5 6

When *n* is 4, The Output is

1

2 3

4 5 6

7 8 9 10

**Task 5:**

Input an integer *n*, output *n\*n* matrix made of numbers 1~*n*.

For example, when *n* is 3, The output is

1 2 3

2 3 1

3 1 2

When *n* is 4, The Output is

1 2 3 4

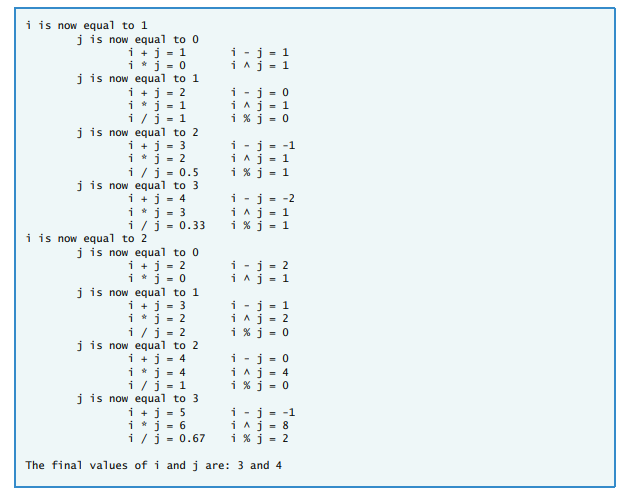
2 3 4 1

3 4 1 2

4 1 2 3

**Exercise 3: Debugging. The program in this section does not run properly. Fix all the compilation errors so that the program will compile successfully. Once the program compiles, compare the output with the sample output, and eliminate any logic errors that may exist. The sample output demonstrates what the program’s output should be once the program’s code has been corrected.**

**Sample Output**



**Broken Code**

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

int i = 1;

double a;

double b;

cout << setprecision( 2 );

for ( int i; i <= 2; i++ )

cout << "i is now equal to " << i << endl;

for ( int j; j <= 3; j++ )

{

cout << "\tj is now equal to " << j << endl;

cout << "\t\ti + j = " << i + j << "\ti - j = "

<< i - j << endl;

cout << "\t\ti \* j = " << i \* j << "\ti ^ j = "

<< pow( i, j ) << endl;

if ( j = 0 )

continue;

else

{

a = i;

b = j;

cout << "\t\ti / j = " << a / b

"\ti % j = " << a % b << endl;

} // end else

} // end for

cout << "\nThe final values of i and j are: " << i

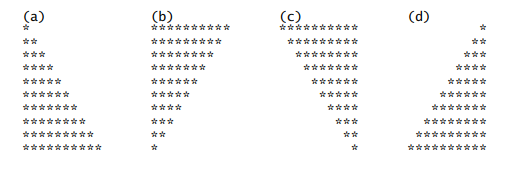
<< " and " << j << endl;

} // end main

**Appendix: Lab Instruction**

Write a program that uses for statements to print the following patterns separately, one below the other. Use for loops to generate the patterns. All asterisks (\*) should be printed by a single statement of the form cout <<'\*'; (this causes the asterisks to print side by side). [Hint: The last two patterns require that each line begin with

an appropriate number of blanks.]



**Template**

#include <iostream>

using namespace std;

int main()

{

int row; // the row position

int column; // the column position

int space; // number of spaces to print

// first triangle

/\* Write a for header to iterate row from 1 to 10 \*/

{

/\* Write a for header to iterate column from 1 to row \*/

cout << "\*";

cout << endl;

} // end for

cout << endl;

// second triangle

/\* Write a for header to iterate row from 10 down to 1 \*/

{

/\* Write a for header to iterate column from 1 to row \*/

cout << "\*";

cout << endl;

} // end for

cout << endl;

// third triangle

/\* Write a for header to iterate row from 10 down to 1 \*/

{

/\* Write a for header to iterate space from 10 down to one more than row \*/

cout << " ";

/\* Write a for header to iterate column from 1 to row \*/

cout << "\*";

cout << endl;

} // end for

cout << endl;

// fourth triangle

/\* Write a for header to iterate row from 10 down to 1 \*/

{

/\* Write a for header to iterate space from 1 to one less than row \*/

cout << " ";

/\* Write a for header to iterate column from 10 down to row \*/

cout << "\*";

cout << endl;

} // end for

} // end main

**Solution for first triangle （If you still have questions with the template, get some hints here. It would be better if you didn't need it.）**

// first triangle

for ( row = 1; row <= 10; row++ )

{

for ( column = 1; column <= row; column++ )

cout << "\*";

cout << endl;

} // end for