软件工程 一班 马启凡 202430551135 **实验名称：循环和数组**

**实验内容：**

**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

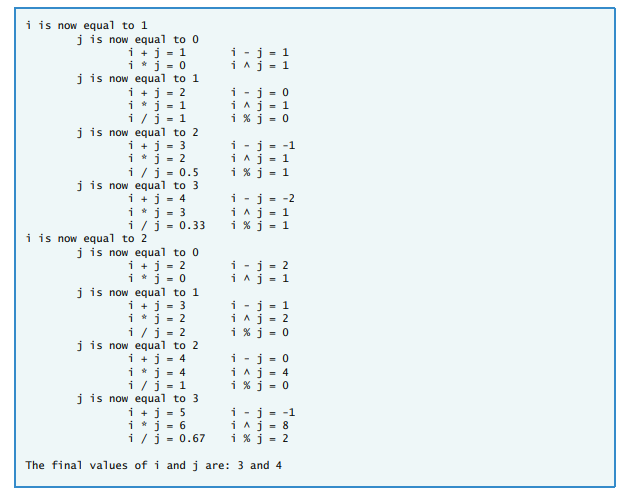
2 3 4 1

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

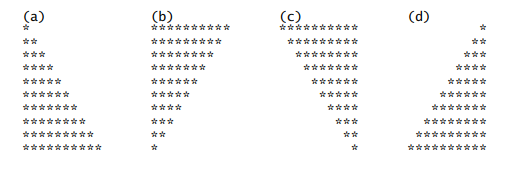
<< " and " << j << endl;

} // 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

**实验过程**

1. **代码实现**

**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 \*\*\*\*.

int main()

{

int n;

cin >> n;

for (int i = 0; i < n; i++)

{

cout << "\*";

}

cout << endl;

system("pause");

return 0;

}

该任务较为简单。为后面任务的铺垫

**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

\*\*\*\*

\*\*\*\*

\*\*\*\*

\*\*\*\*

int main()

{

int n;

cin >> n;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

cout << "\*";

}

cout << endl;

}

system("pause");

return 0;

}

该任务较为简单。为后面任务的铺垫

**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)

int main()

{

int n;

cin >> n;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < i+1; j++)

{

cout << "\*";

}

cout << endl;

}

system("pause");

return 0;

}

该任务较为简单。为后面任务的铺垫

**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

\*

\*\*\*

\*\*\*\*\*

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

int main()

{

int n;

cin >> n;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n-i-1; j++)

{

cout << " ";

}

for (int k = 0; k < 2\*i+1; k++)

{

cout << "\*";

}

cout << endl;

}

system("pause");

return 0;

}

该任务较为简单。为后面任务的铺垫

主要是要找到每一行与该行拥有的星号及星号位置的关系

**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.

int main()

{

int n;

cin >> n;

for (int i = 0; i < n; i++)

{

cout << i+1 << " ";

}

cout << endl;

system("pause");

return 0;

}

该任务较为简单。为后面任务的铺垫

**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

int main()

{

int n;

cin >> n;

for (int i = 0; i < n; i++)

{

for (int j= 0; j < n; j++)

{

cout << j + 1 << " ";

}

cout << endl;

}

cout << endl;

system("pause");

return 0;

}

该任务较为简单。为后面任务的铺垫

**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

int main()

{

int n;

int counter = 1;

cin >> n;

for (int i = 0; i < n; i++)

{

for (int j= 0; j < n; j++)

{

cout << counter << " ";

counter++;

}

cout << endl;

}

cout << endl;

system("pause");

return 0;

}

该任务核心在于设置一个在for循环外的变量进行计数，从而实现数字的累加

**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

int main()

{

int n;

int counter = 1;

cin >> n;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n - i - 1; j++)

{

cout << " ";

}

for (int k = 0; k < i + 1; k++)

{

cout << counter<<" ";

counter++;

}

cout << endl;

}

system("pause");

return 0;

}

该任务和之前的星号有异曲同工之妙，是一样的

**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

int main()

{

int n;

cin >> n;

for (int i = 0; i < n; i++)

{

for (int j = i+1; j <= n; j++)

{

cout << j<<" ";

}

for (int k = 1; k <= i; k++)

{

cout << k<<" ";

}

cout << endl;

}

system("pause");

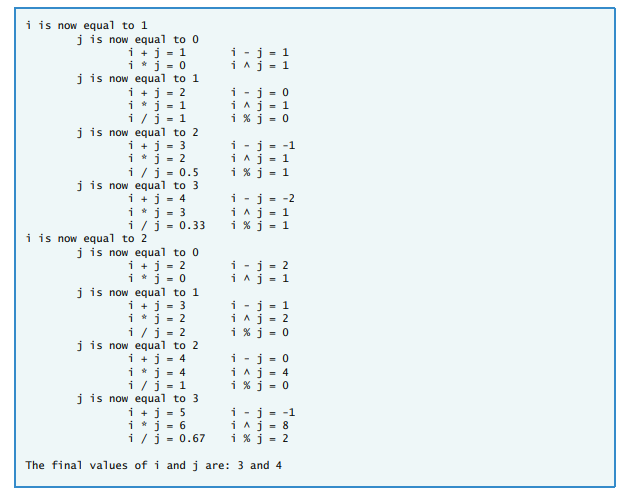
return 0;

}

该代码关键在于实现依次轮换，根据行数进行倒序

**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

//修改后的代码

#include <iostream>

#include <iomanip>

#include<cmath>

using namespace std;

int main()

{

int i = 1;

int j = 0;

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

{

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

j = 0;

for (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 = "<<

fixed << setprecision(0) << pow(i, j) << endl;

if (j ==0)

{

continue;

}

else

{

double a = i;

double b = j;

if (i%j == 0)

{

cout << "\t\ti / j = " << fixed << setprecision(0) << a / b;

cout << "\ti % j = " << i % j << endl;

}

else

{

cout << "\t\ti / j = " << fixed << setprecision(2) << a / b;

cout << "\ti % j = " << i % j << endl;

}

} // end else

} // end for

}

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

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

system("pause");

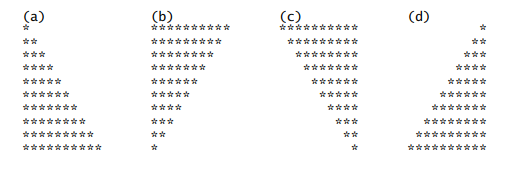
return 0;

} // 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

//代码

#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 \*/

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

{

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

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

{

cout << "\*";

}

cout << endl;

} // end for

cout << endl;

// second triangle

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

for (row = 10; row >= 1; row--)

{

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

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

{

cout << "\*";

}

cout << endl;

} // end for

cout << endl;

// third triangle

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

{

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

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

{

cout << " ";

}

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

for (column = 10; column >= row; column--)

{

cout << "\*";

}

cout << endl;

} // end for

cout << endl;

// fourth triangle

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

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

{

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

for (space = 10; space > row; space--)

{

cout << " ";

}

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

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

{

cout << "\*";

}

cout << endl;

} // end for

system("pause");

return 0;

} // end main

该任务重点在于区分不同类型的三角形对应的空格与星号的特点，再根据for循环的嵌套进行实现

**实验总结**

通过上述的例题进行了一系列循环结构的使用，我重点了解了循环结构各自的特点，尤其是for类型的不同特性，了解了循环结构中容易出现的语法与逻辑错误，在不同题型的对照中意识到循环结构对于程序设计以及运行的重要性。