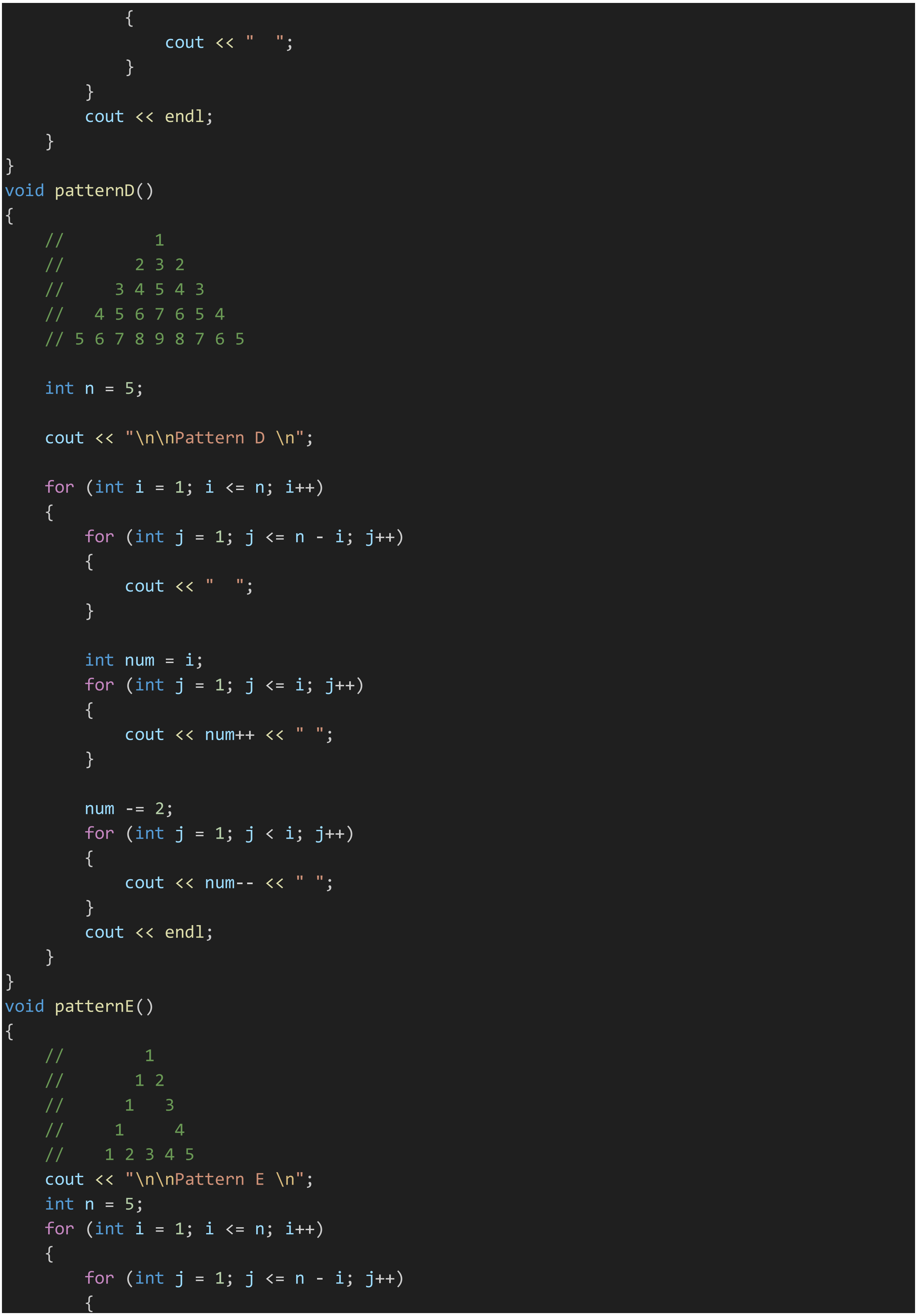
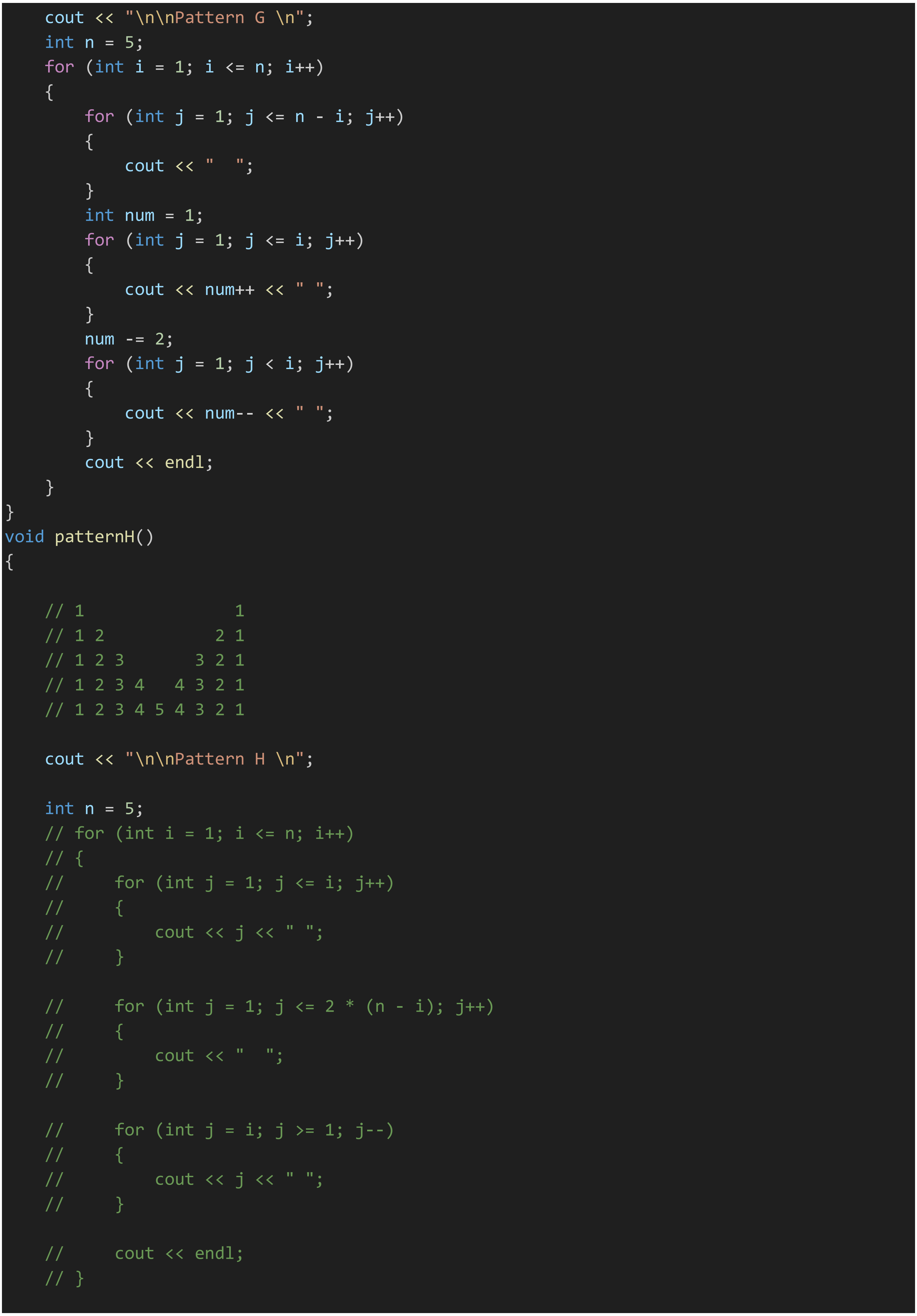
Assignment 02

```
#include <bits/stdc++.h>
using namespace std;
void patternA()
{
    //     *
    //    * *
    //   *   *
    //  *     *
    // *       *
    // *       *
    //  *     *
    //   *   *
    //    * *
    //     *
    cout << "\n\nPattern A \n";
    int n = 5;
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= n - i; j++)
        {
            cout << " ";
        }
        for (int j = 1; j <= i; j++)
        {
            if (j == i || j == 1)
            {
                cout << "* ";
            }
            else
            {
                cout << "  ";
            }
        }
        cout << endl;
    }
    for (int i = n - 1; i >= 1; i--)
    {
        for (int j = 1; j <= n - i; j++)
        {
            cout << " ";
        }
        for (int j = 1; j <= i; j++)
        {
            if (j == i || j == 1)
            {
                cout << "* ";
            }
            else
            {
                cout << "  ";
            }
        }
        cout << endl;
    }
}
void patternB()
{
    // *
    // * *
    // * * *
    // * * * *
    // * * * * *
    // * * * *
    // * * *
    // * *
    // *
    int n = 5;
    cout << "\n\nPattern B \n";
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= i; j++)
        {
            cout << "* ";
        }
        cout << endl;
    }
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= n - i; j++)
        {
            cout << "* ";
        }
        cout << endl;
    }
}
void patternC()
{
    // 1
    // 1 2
    // 1   3
    // 1     4
    // 1 2 3 4 5
    int n = 5;
    cout << "\n\nPattern C \n";
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= i; j++)
        {
            if (j == 1 || j == i || i == n)
            {
                cout << j << " ";
            }
            else
```



```
            {
                cout << "  ";
            }
        }
        cout << endl;
    }
}
void patternD()
{
    //         1
    //       2 3 2
    //     3 4 5 4 3
    //   4 5 6 7 6 5 4
    // 5 6 7 8 9 8 7 6 5

    int n = 5;

    cout << "\n\nPattern D \n";

    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= n - i; j++)
        {
            cout << "  ";
        }

        int num = i;
        for (int j = 1; j <= i; j++)
        {
            cout << num++ << " ";
        }

        num -= 2;
        for (int j = 1; j < i; j++)
        {
            cout << num-- << " ";
        }
        cout << endl;
    }
}
void patternE()
{
    //        1
    //       1 2
    //      1   3
    //     1     4
    //    1 2 3 4 5
    cout << "\n\nPattern E \n";
    int n = 5;
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= n - i; j++)
        {
            cout << " ";
        }
        for (int j = 1; j <= n; j++)
        {
            if (i == j || j == 1 || i == n)
            {
                cout << j << " ";
            }
            else
            {
                cout << "  ";
            }
        }
        cout << endl;
    }
}
void patternF()
{
    // 1 2 3 4 5
    // 2     5
    // 3   5
    // 4 5
    // 5

    cout << "\n\nPattern F \n";

    int n = 5;
    for (int i = 1; i <= n; i++)
    {

        int num = i;
        for (int j = 1; j <= n; j++, num++)
        {
            if (j == ((n - i) + 1) || j == 1 || i == 1)
            {
                cout << num << " ";
            }
            else
            {
                cout << "  ";
            }
        }
        cout << endl;
    }
}
void patternG()
{
    //         1
    //       1 2 1
    //     1 2 3 2 1
    //   1 2 3 4 3 2 1
    // 1 2 3 4 5 4 3 2 1
```



```
    cout << "\n\nPattern G \n";
    int n = 5;
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= n - i; j++)
        {
            cout << "  ";
        }
        int num = 1;
        for (int j = 1; j <= i; j++)
        {
            cout << num++ << " ";
        }
        num -= 2;
        for (int j = 1; j < i; j++)
        {
            cout << num-- << " ";
        }
        cout << endl;
    }
}
void patternH()
{

    // 1               1
    // 1 2           2 1
    // 1 2 3       3 2 1
    // 1 2 3 4   4 3 2 1
    // 1 2 3 4 5 4 3 2 1

    cout << "\n\nPattern H \n";

    int n = 5;
    // for (int i = 1; i <= n; i++)
    // {
    //     for (int j = 1; j <= i; j++)
    //     {
    //         cout << j << " ";
    //     }

    //     for (int j = 1; j <= 2 * (n - i); j++)
    //     {
    //         cout << "  ";
    //     }

    //     for (int j = i; j >= 1; j--)
    //     {
    //         cout << j << " ";
    //     }

    //     cout << endl;
    // }
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= i; j++)
        {
            cout << j << " ";
        }

        int spaces = 2 * (n - i);
        for (int k = 0; k < spaces; k++)
        {
            cout << "  ";
        }

        for (int j = i; j >= 1; j--)
        {
            cout << j << " ";
        }

        cout << endl;
    }
}

int main()
{
    patternA();
    patternB();
    patternC();
    patternD();
    patternE();
    patternF();
    patternG();
    patternH();
    return 0;
}
```

Output:

```
PS D:\Fullstack-Java-FirstBit-Solutions> cd 'd:\Fullstack-Java-FirstBit-Solutions\DSA\Assignments\Patterns\Assignment02\output'

PS D:\Fullstack-Java-FirstBit-Solutions\DSA\Assignments\Patterns\Assignment02\output> & .\'q1.exe'
```

Pattern A

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

Pattern B

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

Pattern C

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

Pattern D

```
        1
      2 3 2
    3 4 5 4 3
  4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5

Pattern E
    1
   1 2
  1   3
 1     4
1 2 3 4 5

Pattern F
1 2 3 4 5
2     5
3   5
4 5
5

Pattern G
        1
      1 2 1
    1 2 3 2 1
  1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1

Pattern H
1                 1
1 2             2 1
1 2 3         3 2 1
1 2 3 4     4 3 2 1
1 2 3 4 5 5 4 3 2 1
PS D:\Fullstack-Java-FirstBit-Solutions\DSA\Assignments\Patterns\Assignment02\output>
```