Task 1:

1. Create a class *CVector* which represents a vector of integers. The exact size of a vector is unknown; therefore memory has to be reserved dynamically. The constructor should have the following form: *CVector(unsigned el);* where *el* is the actual number of elements to be included in the vector; initially, the vector has to be filled in with zeros. Redefine the operator [] to allow for reading and writing of individual vector elements.

2. Construct a class *CMatrix*. A matrix should be composed of a number of vectors representing rows. The constructor of the class receives 2 arguments (the number of rows and the number of columns) and fills in the matrix with zeros. Redefine the operator * (multiplication). This is defined in the following manner:

For two matrices **A**(m×n) and **B**(n×l) the result of multiplication is a new matrix **X**(m×l) with the following elements:

$$x_i^j = \sum_{k=0}^{n-1} a_i^k * b_k^j, \quad i = 1,...,m; \quad j = 1,...,l$$

3. Construct a class *CBooleanMatrix* which represents a Boolean matrix. Redefine the operator * (multiplication). This is defined in the following manner:

For two Boolean matrices **A**(m×n) and **B**(n×l) the result of multiplication is a new Boolean matrix **X**(m×l) with the following elements:

$$x_i^j = \bigvee_{k=0}^{n-1} a_i^k \wedge b_k^j, \quad i = 1,...,m; \quad j = 1,...,l$$

4. Implement the following *main* function:

```
int main()
{
      using namespace std;

      CMatrix m1(3, 2);
      m1[0][0] = 2;     m1[0][1] = 3;
      m1[1][0] = 4;     m1[1][1] = 5;
      m1[2][0] = 1;     m1[2][1] = 4;

      CMatrix m2(2, 4);
      m2[0][0] = 1;   m2[0][1] = 2;   m2[0][2] = 3;   m2[0][3] = 4;
      m2[1][0] = 4;   m2[1][1] = 3;   m2[1][2] = 2;   m2[1][3] = 1;

      cout << m1 << endl;
      cout << m2 << endl;
      cout << m1 * m2 << endl;

      CBooleanMatrix mb1(3, 2);
      mb1[0][0] = mb1[1][1] = mb1[2][0] = mb1[2][1] = 1;
      CBooleanMatrix mb2(2, 4);
      mb2[0][2] = mb2[0][3] = mb2[1][1] = mb2[1][3] = 1;

      cout << mb1 << endl;
      cout << mb2 << endl;
      cout << mb1 * mb2 << endl;

      cout << (mb1 = mb2) << endl;

      return 0;
}
```

5. The results should be the following:

```
    2    3
    4    5
    1    4


    1    2    3    4
    4    3    2    1


   14   13   12   11
   24   23   22   21
   17   14   11    8


    1    0
    0    1
    1    1


    0    0    1    1
    0    1    0    1


    0    0    1    1
    0    1    0    1
    0    1    1    1


    0    0    1    1
    0    1    0    1
```

Task 2:

Check if the following code is correct. What is the output of this program?

```
class A
{
      int i;
public:
      A (int i ) { this->i = i; cout << "constructor" << endl;}
      ~A () { cout << "destructor" << endl; }
      operator int () const { return i; }
};

int test (int k)
{
      static A a;
      return a * k;
}

int main()
{
      cout << test(3) << endl;
      cout << test(4) << endl;
      cout << test(5) << endl;
}
```