

Есть код, реализующий некоторый функционал для класса Matrix в файле «TestMatrix.h»:

```
#include <iostream>

template <typename T>
class Matrix {
private:
    T** data;
    size_t rows, columns;

public:
    Matrix(size_t m, size_t n): rows(m), columns(n) {
        data = new T * [rows];
        size_t i = 0;
        try {
            for (; i != rows; ++i) {
                data[i] = new T[columns];
            }
        } catch (...) {
            for (size_t k = 0; k != i; ++k) {
                delete [] data[k];
            }
            delete [] data;
            throw;
        }
    }

    T* operator [] (size_t i) {
        return data[i];
    }
    const T* operator [] (size_t i) const {
        return data[i];
    }

    size_t GetRows() const {
        return rows;
    }

    size_t GetColumns() const {
        return columns;
    }

    ~Matrix() {
        for (size_t i = 0; i != rows; ++i) {
            delete [] data[i];
        }
        delete [] data;
    }

    // ADD YOUR CODE HERE
};

template <typename T>
Matrix<T> FillMatrix(size_t m, size_t n) {
```

```

Matrix<T> A(m, n);
for (size_t i = 0; i != m; ++i) {
    for (size_t j = 0; j != n; ++j) {
        A[i][j] = i + j;
    }
}
return A;
}

template <typename T>
std::ostream& operator << (std::ostream& out, const Matrix<T>& A) {
    for (size_t i = 0; i != A.GetRows(); ++i) {
        for (size_t j = 0; j != A.GetColumns(); ++j) {
            out << A[i][j] << " ";
        }
        out << "\n";
    }
    return out;
}

```

При этом следующий пример использования класса Matrix не работает!

```

#include "TestMatrix.h"
#include <iostream>

int main() {
    size_t m, n;
    std::cin >> m >> n;
    Matrix<int> A(m, n);
    // ...
    A = FillMatrix<int>(m, n);
    std::cout << A << "\n";
}

```

Объясните, почему не работает код выше и допишите код (см. строку // ADD YOUR CODE HERE) так, чтобы пример заработал. Менять уже имеющийся код нельзя, можно только добавить свой.