

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АДЫГЕЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Инженерно-физический факультет
Кафедра автоматизированных систем обработки информации и
управления

ОТЧЕТ ПО ПРАКТИКЕ

Вариант 7

Нахождение определителя матрицы

1 курс, группа 1ИВТ АСОИУ

Выполнил:

_____ Г. К. Арустамов
«___» _____ 2024 г.

Руководитель:

_____ С. В. Теплоухов
«___» _____ 2024 г.

Майкоп, 2024 г.

Содержание

- 1) Задача
- 2) Пример кода, решающего данную задачу
- 3) Скриншот работы программы

1. Задача

Найти определитель матрицы.

2. Пример кода

```
#include <iostream>
```

```
#include <cmath>
```

```
using namespace std;
```

```
//Данная программа вычисляет определитель  
матрицы используя рекурсию
```

```
void clearMemory(int** a, int n) { //Функция  
освобождения памяти, выделенной под  
двумерный динамический массив  
for (int i = 0; i < n; i++) {
```

```

    delete[] a[i];
}
delete[] a;
}

```

```

int findDet(int** a, int n) { //Рекурсивная
    функция вычисления определителя матрицы
    if (n == 1)
        return a[0][0];
    else if (n == 2)
        return a[0][0] * a[1][1] - a[0][1] * a[1][0];
    else {
        int d = 0;
        for (int k = 0; k < n; k++) {
            int** m = new int* [n - 1];
            for (int i = 0; i < n - 1; i++) {
                m[i] = new int[n - 1];
            }
            for (int i = 1; i < n; i++) {
                int t = 0;
                for (int j = 0; j < n; j++) {
                    if (j == k)
                        continue;
                    m[i - 1][t] = a[i][j];
                    t++;
                }
            }
        }
    }
}

```

```

    }

    }
    d += pow(-1, k + 2) * a[0][k] *
    findDet(m, n - 1);
    clearMemory(m, n - 1); //Освобождаем память,
    выделенную под алгебраическое дополнение
    }

    return d; //Возвращаем определитель матрицы
}
}

int main() {
    int n; //Объявляем целочисленную переменную
    cout << "Enter a matrix size:\n";
    cout << "n = ";

    cin >> n; //Вводим размерность матрицы
    int** a = new int* [n]; //Объявляем двумерный
    целочисленный динамический массив (матрицу)
    for (int i = 0; i < n; i++) {
        a[i] = new int[n];

    }
}

```

```

cout << "Enter a matrix:\n";
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        cin >> a[i][j]; //Вводим элементы матрицы
    }

}
cout << "Found determinant: " << findDet(a, n)
    << "\n"; //Вызываем рекурсивную функцию
    вычисления определителя матрицы

clearMemory(a, n); //Освобождаем память,
    выделенную под исходную матрицу

system("pause");
return 0;

}

```

3. Скриншот работы программы

```
Enter a matrix size:  
n = 2  
Enter a matrix:  
1  
2  
3  
4  
Found determinant: -2  
Для продолжения нажмите любую клавишу . . .
```

Рис. 1. Результат