Отчёт по лабораторной работе №9 «Указатели. Динамическое выделение памяти»

# Вариант №1 (язык C)

## Титульный лист

Лабораторная работа №9 «Указатели. Динамическое выделение памяти»  
Студент: \_\_\_\_\_\_\_\_\_\_\_\_  
Группа: \_\_\_\_\_\_\_\_\_\_\_\_  
Дата выполнения: \_\_\_\_\_\_\_\_\_\_\_\_

## Цель работы

Приобрести навыки работы с указателями и динамическим выделением памяти.

## Текст задания

Переделать программу из лабораторной работы №7 (вариант 1), используя динамическое выделение и освобождение памяти для всех используемых массивов. Реализация на языке C.

## Назначение идентификаторов (переменных и функций)

- matrix – двумерный массив, реализованный через указатели.  
- rows, cols – размеры матрицы.  
- countNonZeroRows – подсчёт строк без нулей.  
- findMaxRepeatingElement – поиск максимального повторяющегося элемента.  
- Освобождение памяти через free().

## Программный код

#include <stdio.h>  
#include <stdlib.h>  
#include <limits.h>  
  
int countNonZeroRows(int \*\*matrix, int rows, int cols) {  
 int count = 0;  
 for (int i = 0; i < rows; i++) {  
 int zero\_found = 0;  
 for (int j = 0; j < cols; j++) {  
 if (matrix[i][j] == 0) {  
 zero\_found = 1;  
 break;  
 }  
 }  
 if (!zero\_found)  
 count++;  
 }  
 return count;  
}  
  
int findMaxRepeatingElement(int \*\*matrix, int rows, int cols) {  
 int freq[201] = {0};  
 int max\_repeat = INT\_MIN;  
 for (int i = 0; i < rows; i++)  
 for (int j = 0; j < cols; j++)  
 freq[matrix[i][j] + 100]++;  
  
 for (int i = 0; i < 201; i++) {  
 if (freq[i] > 1 && (i - 100) > max\_repeat)  
 max\_repeat = i - 100;  
 }  
 return max\_repeat;  
}  
  
int main() {  
 int rows, cols;  
 printf("Введите количество строк и столбцов: ");  
 scanf("%d %d", &rows, &cols);  
  
 int \*\*matrix = (int \*\*)malloc(rows \* sizeof(int \*));  
 for (int i = 0; i < rows; i++)  
 matrix[i] = (int \*)malloc(cols \* sizeof(int));  
  
 printf("Введите элементы матрицы:");  
 for (int i = 0; i < rows; i++)  
 for (int j = 0; j < cols; j++)  
 scanf("%d", &matrix[i][j]);  
  
 int count = countNonZeroRows(matrix, rows, cols);  
 printf("Количество строк без нулей: %d  
", count);  
  
 int max\_repeat = findMaxRepeatingElement(matrix, rows, cols);  
 if (max\_repeat != INT\_MIN)  
 printf("Максимальный повторяющийся элемент: %d", max\_repeat);  
 else  
 printf("Повторяющиеся элементы отсутствуют.");  
  
 for (int i = 0; i < rows; i++)  
 free(matrix[i]);  
 free(matrix);  
  
 return 0;  
}

## Результаты вычисления

Изображение выглядит как текст, Шрифт, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.