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

# Вариант №13 (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 13), используя динамическое выделение и освобождение памяти для всех используемых массивов. Реализация на языке C.

## Назначение идентификаторов (переменных и функций)

- matrix – указатель на двумерный массив.  
- size – размер квадратной матрицы.  
- zeroNonDiagonalElements – обнуляет элементы вне главной диагонали.  
- changeSignMaxAbsInRow – меняет знак максимального по модулю элемента в каждой строке.  
- Освобождение памяти через free().

## Программный код

#include <stdio.h>  
#include <stdlib.h>  
#include <math.h>  
  
void zeroNonDiagonalElements(int \*\*matrix, int size) {  
 for(int i = 0; i < size; i++) {  
 for(int j = 0; j < size; j++) {  
 if(i != j) matrix[i][j] = 0;  
 }  
 }  
}  
  
void changeSignMaxAbsInRow(int \*\*matrix, int size) {  
 for(int i = 0; i < size; i++) {  
 int max\_idx = 0;  
 for(int j = 1; j < size; j++) {  
 if(abs(matrix[i][j]) > abs(matrix[i][max\_idx])) {  
 max\_idx = j;  
 }  
 }  
 matrix[i][max\_idx] = -matrix[i][max\_idx];  
 }  
}  
  
int main() {  
 int size;  
 printf("Введите размер матрицы: ");  
 scanf("%d", &size);  
  
 int \*\*matrix = (int \*\*)malloc(size \* sizeof(int \*));  
 for(int i = 0; i < size; i++) {  
 matrix[i] = (int \*)malloc(size \* sizeof(int));  
 }  
  
 printf("Введите элементы матрицы:\n");  
 for(int i = 0; i < size; i++) {  
 for(int j = 0; j < size; j++) {  
 scanf("%d", &matrix[i][j]);  
 }  
 }  
  
 changeSignMaxAbsInRow(matrix, size);  
 zeroNonDiagonalElements(matrix, size);  
  
 printf("Результирующая матрица:\n");  
 for(int i = 0; i < size; i++) {  
 for(int j = 0; j < size; j++) {  
 printf("%d ", matrix[i][j]);  
 }  
 printf("\n");  
 }  
  
 for(int i = 0; i < size; i++) {  
 free(matrix[i]);  
 }  
 free(matrix);  
  
 return 0;  
}

## Результаты вычисления

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.