

# Лабораторная работа №5

## Задачи

**Выполнил:**

*Студент 1-го курса*

*Группы ИВТ-1.1*

*Шардт Максим Александрович*

### 1. Многофайловый проект, условная компиляция, утилита Make

1.1. Напишите программу из нескольких файлов (модулей), включая файл основной программы. Файлы должны содержать вынесенные отдельно функции для выделения памяти под динамические двумерные и одномерные массивы и функции для перемножения матриц. Собрать проект используя утилиту Make.

*Код программы*

Определения функций

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <time.h>
4
5  double **alloc_array(int block_count)
6  {
7      return (double **)malloc(sizeof(double) * block_count);
8  }
9
10 double **alloc_matrix(int n_rows, int n_columns)
11 {
12     double **matrix = (double **)malloc(n_rows * sizeof(double));
13
14     for (int i = 0; i < n_rows; i++)
15         matrix[i] = (double *)malloc(n_columns * sizeof(double));
16
17     return matrix;
18 }
19
20 void fill_matrix(double **matrix, int n_rows, int n_columns)
21 {
22     srand(time(NULL));
23     for (int i = 0; i < n_rows; i++)
```

```

24     for (int j = 0; j < n_columns; j++)
25         matrix[i][j] = random() % 101;
26 }
27
28 void print_matrix(double **matrix, int n_rows, int n_columns)
29 {
30     printf("Matrix:\n");
31     for (int i = 0; i < n_rows; i++)
32     {
33         for (int j = 0; j < n_columns; j++)
34             printf("%3.0lf ", matrix[i][j]);
35         printf("\n");
36     }
37 }
38 void free_matrix(void **matrix, int n_rows)
39 {
40     for (int i = 0; i < n_rows; i++)
41         free(matrix[i]);
42     free(matrix);
43 }
44
45 void multiply_matrix(int M, int N, int K, double **target, double
**matrix_a, double **matrix_b)
46 {
47     for (int i = 0; i < M; i++)
48     {
49         for (int j = 0; j < N; j++)
50         {
51             target[i][j] = 0;
52             for (int k = 0; k < K; k++)
53                 target[i][j] += matrix_a[i][k] * matrix_b[k][j];
54         }
55     }
56 }

```

*Листинг 1: my\_functions.c*

### Заголовочный файл с объявлениями функций

```

1 #ifndef FUNCTIONS
2 #define FUNCTIONS
3
4 double *alloc_array(int block_count);
5

```

```

6 double **alloc_matrix(int n_rows, int n_columns);
7
8 void fill_matrix(double **matrix, int n_rows, int n_columns);
9
10 void print_matrix(double **matrix, int n_rows, int n_columns);
11
12 void free_matrix(void **matrix, int n_rows);
13
14 void multiply_matrix(int M, int N, int K, double **target, double
15 **matrix_a, double **matrix_b);
16
17 #endif

```

*Листинг 2: my\_functions.h*

## Main

```

1 #include <stdio.h>
2 #include "my_functions.h"
3
4 int main(void)
5 {
6     double **matrix_a = alloc_matrix(3, 3);
7     fill_matrix(matrix_a, 3, 3);
8     double **matrix_b = alloc_matrix(3, 3);
9     fill_matrix(matrix_b, 3, 3);
10    double **matrix_c = alloc_matrix(3, 3);
11    multiply_matrix(3, 3, 3, matrix_c, matrix_a, matrix_b);
12
13    print_matrix(matrix_a, 3, 3);
14    print_matrix(matrix_b, 3, 3);
15    print_matrix(matrix_c, 3, 3);
16
17    return 0;
18 }

```

*Листинг 3: main.c*

## Makefile

```

1 all:
2     gcc main.c my_functions.c -o output
3
4 debug:
5     gdb main.c my_functions.c

```

*Результат выполненной работы*

```
Matrix:
 12  22  61
 47  19  60
 35  64  12
Matrix:
 83  67  74
 95  97  56
 40 100  88
Matrix:
5526 9038 7488
8106 10992 9822
9465 9753 7230
```