

Министерство науки и образования РФ
Федеральное государственное автономное образовательное
учреждение высшего профессионального образования
«Санкт-Петербургский государственный электротехнический
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»
(СПбГЭТУ «ЛЭТИ»)
Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

Отчет по лабораторной работе №4
на тему:
“Функции и указатели в языке C/C++”
по дисциплине “Программирование”

Выполнил: Табаков А.В.

Группа 4306

Принял: к.т.н., доцент Сискович Т.И.

Санкт-Петербург
2014 г.

Цель

Получить практические навыки работы с функциями и указателями на языке «C/C++».

Задание

Разработать алгоритм и написать программу, где необходимо создать новую матрицу из столбцов исходной, элементы которой расположены по возрастанию. Программа должна содержать меню с выбором действий и выполняться многократно, по желанию пользователя.

Уточнение задания

Вначале программа должна предложить задать размер матрицы (mas1). Под матрицу должна выделяться динамическая память. Необходимо вывести меню. Если пользователь выберет несуществующий пункт меню, ему будет выведено сообщение об ошибке.

Пункты меню:

- 1: Ввод матрицы.
- 2: Вывод введенной матрицы.
- 3: Обработка исходной матрицы
- 4: Вывод новой матрицы.
- 5: Выход.

Контрольные примеры

Контрольные примеры представлены в Таблице 1.

Таблица 1. Контрольные примеры

№ примера	Исходные данные			Результат
	rows	col	mas1	
1	4	0	-	Кол-во элементов (целые положительные числа) Введите кол-во повторно!
2	3	4	3 -3 -2 0 3 0 -4 -2 -5 -4 -3 -5	Результат не может быть сформирован!
3	2	3	4 1 4 3	4 1 4 3
4	2	2	-3 3 -2 -4	-3 -2
5	3	3	3 -2 -5 -3 -3 -3 -4 2 0	-5 -3 0

Описание главной функции

Описание переменных главной функции

Описание переменных представлено в Таблице 2.

Таблица 2. Описание переменных главной функции

Имя переменной	Тип	Назначение
mas1	int**	Указатель на адрес первого элемента исходной матрицы
rows	int	Количество строк матрицы
col	int	Количество столбцов матрицы
mas2	int**	Указатель на адрес первого элемента сформированной матрицы
ncol	int	Количество столбцов сформированной матрицы
check	int	Переменная проверки наличия введенной матрицы
check_2	int	Переменная проверки обработки матрицы
Q	int	Переменная выбора меню

Краткое описание алгоритма

Начало программы.

Шаг №1. Вывод меню.

Шаг №2. Выбор пользователем пункта меню.

Шаг №3. Переход к пункту, выбранным пользователем.

Пункт 1: Ввод матрицы. Переход к шагу 2.

Пункт 2: Вывод введенной матрицы. Переход к шагу 2.

Пункт 3: Формирование новой матрицы из столбцов исходной, элементы которой расположены по возрастанию. Переход к шагу 2.

Пункт 4: Вывод обработанной матрицы. Переход к шагу 2.

Пункт 5: Выход. Переход к шагу 4.

Шаг №4. Конец программы.

Блок-схема главной функции

Блок-схема главной функции представлена на рисунке 1

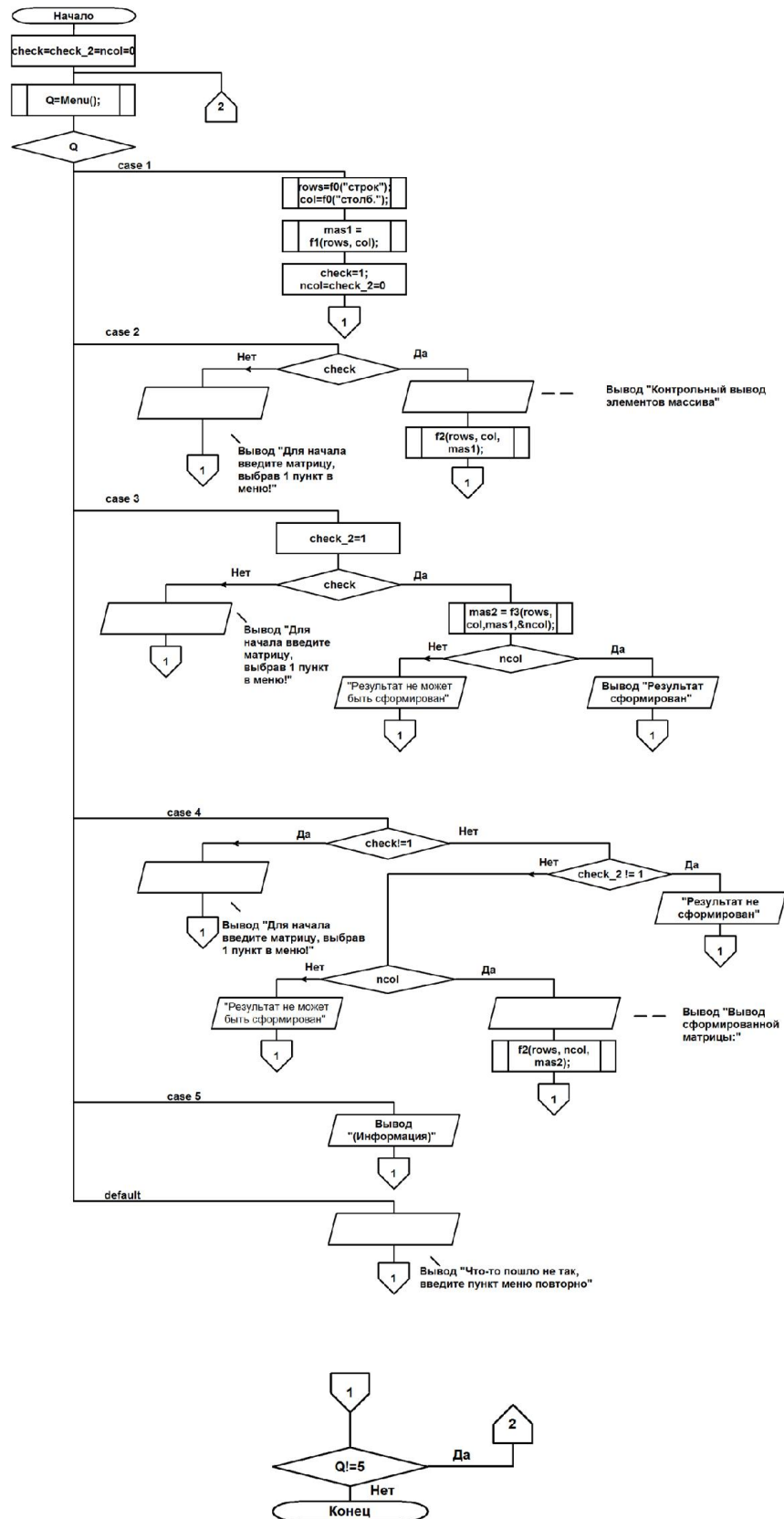


Рис. 1. Блок-схема главной функции

Описание функций

Описание функции f2

Назначение

Функция используется для вывода матрицы.

Прототип

`void f2(int, int, int**);`, где `int**` - указатель на адрес объекта типа `int`, `void` - тип не возвращаемого значения.

Пример вызова

`f2(rows, col, mas1);`, где `rows` – кол-во строк матрицы, `col` – кол-во столбцов матрицы, `mas1` – указатель на адрес первого элемента матрицы.

Описание переменных функции f2

Описание локальных переменных функции `f2` представлены в Таблице 3.

Таблица 3. Описание локальных переменных функции `f2`

Имя переменной	Тип	Назначение
i	int	Номер строки
j	int	Номер столбца

Блок-схема функции f2

Блок-схема функции `f2` представлена на рисунке 2

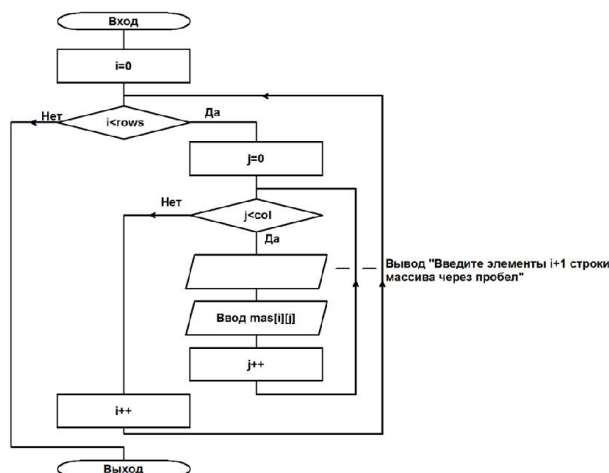


Рис. 2. Блок-схема функции `f2`

Описание функции f0

Назначение

Функция используется для ввода размеров матрицы.

Прототип

int f0(const char*);, где char* - указатель на адрес объекта типа char, int – тип возвращаемого значения.

Пример вызова

rows=f0(“строк”); , где “строк”– строка для формирования правильной подсказки.

Описание переменных функции f0

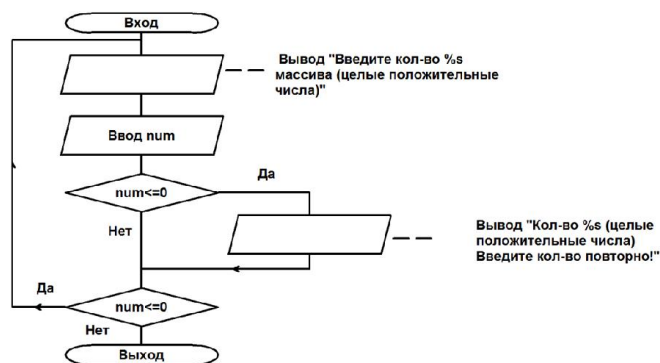
Описание локальных переменных функции f0 представлены в Таблице 4.

Таблица 4. Описание локальных переменных функции f0

Имя переменной	Тип	Назначение
num	int	Количество

Блок-схема функции f0

Блок-схема функции f1 представлена на рисунке 3



Описание функции f1

Назначение

Функция используется для ввода матрицы.

Прототип

`int **f1(int*, int*);`, где `int**` - указатель на адрес объекта типа `int`, `int**` – тип возвращаемого значения.

Пример вызова

`mas1=f1(rows, col);`, где `rows` – кол-во строк матрицы, `col` – кол-во столбцов матрицы.

Описание переменных функции f1

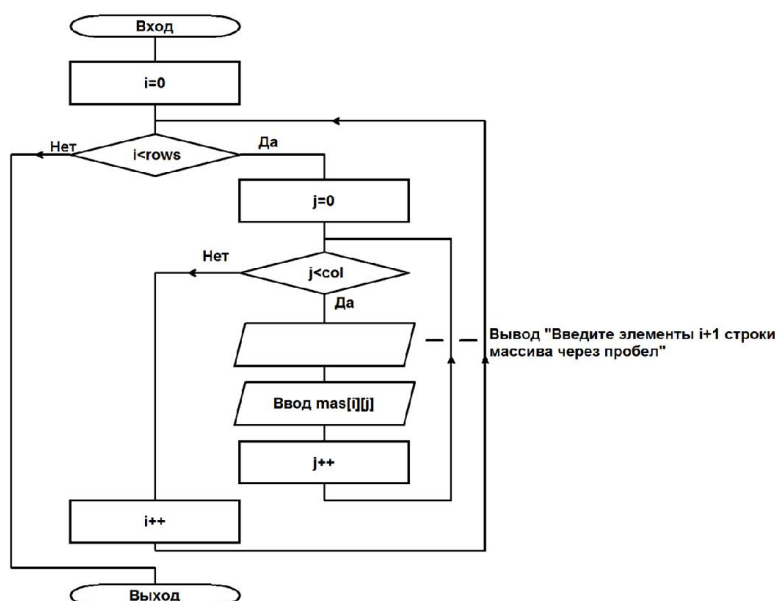
Описание локальных переменных функции `f1` представлены в Таблице 5.

Таблица 5. Описание локальных переменных функции `f1`

Имя переменной	Тип	Назначение
<code>mas1</code>	<code>int**</code>	Указатель на адрес первого элемента исходной матрицы
<code>i</code>	<code>int</code>	Номер строки
<code>j</code>	<code>int</code>	Номер столбца

Блок-схема функции f1

Блок-схема функции `f1` представлена на рисунке 4



Описание функции f3

Назначение

Функция используется для формирования новой матрицы из столбцов исходной, элементы которой расположены по возрастанию.

Прототип

`int **f3(int, int, int**, int*);`, где `int**` - указатель на адрес объекта типа `int`, `int**` – тип возвращаемого значения.

Пример вызова

`mas2=f3(rows,col,mas1,&ncol);`, где `rows` – кол-во строк матрицы, `col` – кол-во столбцов матрицы, `ncol` – указатель на адрес кол-ва столбцов новой матрицы.

Описание переменных функции f3

Описание локальных переменных функции `f3` представлены в Таблице 6.

Таблица 6. Описание локальных переменных функции `f3`

Имя переменной	Тип	Назначение
<code>mas2</code>	<code>int**</code>	Указатель на адрес первого элемента сформированной матрицы
<code>i</code>	<code>int</code>	Номер строки
<code>j</code>	<code>int</code>	Номер столбца
<code>g</code>	<code>int</code>	Кол-во столбцов сформированной матрицы
<code>f</code>	<code>int</code>	Булева переменная

Блок-схема функции f3

Блок-схема функции `f3` представлена на рисунке 5

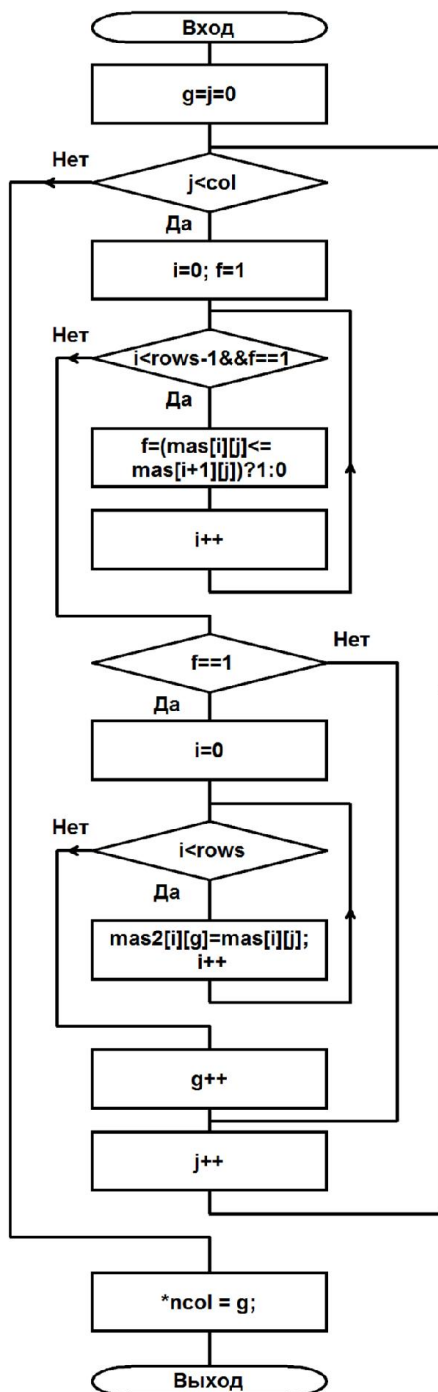


Рис. 5. Блок-схема функции f3

Описание функции fr

Назначение

Функция используется для освобождения памяти, выделенной ранее под матрицы.

Прототип

`void fr(int**, int);`, где `int**` - указатель на адрес объекта типа `int`, `void` - тип не возвращаемого значения.

Пример вызова

`fr(mas1, rows);`, где `mas1` – указатель на адрес первого элемента матрицы, `rows` – кол-во строк матрицы.

Описание переменных функции fr

Описание локальных переменных функции `fr` представлены в Таблице 7.

Таблица 7. Описание локальных переменных функции `fr`

Имя переменной	Тип	Назначение
<code>i</code>	<code>int</code>	Номер строки

Текст программы с комментариями

```
#include <stdio.h>
#include <locale.h>
#include <stdlib.h>
#include <malloc.h>

int Menu(); //Прототип функции меню
int f0(const char*); //Прототип функции ввода размеров матрицы
int **f1(int, int); //Прототип функции ввода матрицы mas1
void f2(int, int, int**); //Прототип функции вывода матрицы
int **f3(int, int, int**, int*); //Прототип функции формирования матрицы mas2
void fr(int**, int); //Прототип функции освобождения памяти

int main(void)
{
    setlocale(LC_ALL, "RUS");
    int **mas1 = NULL, **mas2 = NULL, rows, col, Q, check = 0, check_2 = 0, ncol = 0;
    const char* help_enter = "Для начала введите матрицу, выбрав 1 пункт в меню!\n";
    do
    {
        Q = Menu(); //Меню
        switch (Q)
        {
            case 1: //Ввод матрицы
                system("cls");
```

```

        fr(mas1, rows);                //Освобождение памяти исходной матрицы, если
                                        //есть
        fr(mas2, rows);                //Освобождение памяти сформированной матрицы,
                                        //если есть
        rows = f0("строк");            //Ввод кол-ва строк
        col = f0("столбцов");           //Ввод кол-ва столбцов
        mas1 = f1(rows, col);           //Ввод матрицы
        check = 1;
        ncol = check_2 = 0;
        break;
case 2:                                //Вывод матрицы
    system("cls");
    if (check)
    {
        printf("Контрольный вывод элементов массива\n");
        f2(rows, col, mas1);           //Вывод исходной матрицы
    }
    else
        printf(help_enter);
    system("pause");
    break;
case 3:                                //Обработка матрицы
    system("cls");
    check_2 = 1;
    if (check)
    {
        mas2 = f3(rows, col, mas1, &ncol); //Обработка исходной матрицы
        if (ncol)
            printf("Результат сформирован\n");
        else
            printf("Результат не может быть сформирован\n");
    }
    else
        printf(help_enter);
    system("pause");
    break;
case 4:                                //Вывод сформированной матрицы
    system("cls");
    if (check!=1)
        printf(help_enter);
    else
        if (check_2!=1)
            printf("Результат не сформирован\n");
        else
            if (ncol)
            {
                printf("Вывод сформированной матрицы:\n");
                f2(rows, ncol, mas2);    //Вывод сформированной матрицы
            }
            else
                printf("Результат не может быть сформирован\n");
    system("pause");
    break;
case 5:                                //Выход
    system("cls");
    puts("До новых встреч.");
    puts("Если возникли вопросы или проблемы обращайтесь, пожалуйста, на
        электронную почту: ");
    puts("komdosh@gelezo2.ru");
    system("pause");
    break;
default:
    printf("Что-то пошло не так, введите пункт меню повторно\n");
    system("pause");
}
}

```

```

        while (Q!=5);
        fr(mas1, rows);
        fr(mas2, rows);
        return 0;
    }
}
//*****
//Функция Меню
int Menu()
{
    int Q;
    system("cls");
    puts("Главное меню");
    puts("1 - Ввод матрицы");
    puts("2 - Показать введенную матрицу");
    puts("3 - Обработка исходной матрицы ");
    puts("4 - Показать обработанную матрицу");
    puts("5 - Выход");
    printf("Введите номер пункта - ");
    scanf("%d", &Q);
    printf("\n");
    return Q;
}
//*****
//Функция ввода размеров матрицы
int f0(const char* str)
{
    int num;
    printf("Введите кол-во %s массива (целые положительные числа)\n", str);
    do
    {
        scanf("%d", &num);
        if (num<=0)
            printf("Кол-во %s это целые положительные числа, Введите кол-во повторно!\n", str);
    }
    while (num<=0);
    return num;
}
//*****
//Функция ввода матрицы mas1
int **f1(int rows, int col)
{
    int i, j, **mas;
    mas = (int**) malloc(rows*sizeof(int*));
    for (i = 0; i < rows; i++)
    {
        *(mas+i) = (int*) malloc(col*sizeof(int));
        printf("Введите элементы %d строки массива через пробел\n", i+1);
        for (j = 0; j < col; scanf("%d", &mas[i][j]), j++);
    }
    return mas;
}
//*****
//Функция вывода матрицы
void f2(int rows, int col, int** mas)
{
    for (int i = 0; i<rows; printf("\n"), i++)
        for (int j = 0; j<col; printf("%5d", mas[i][j]), j++);
}
//*****
//Функция формирования матрицы mas2
int **f3(int rows, int col, int** mas, int* ncol)
{
    int i, j, f, g, **mas2 = NULL;
    mas2 = (int**) malloc(rows*sizeof(int*));

    for (i = 0; i<rows; i++)

```

```

        *(mas2+i) = NULL;
    for (g = j = 0; j < col; j++)
    {
        for (i = 0, f = 1; i < rows-1&&f==1; i++)
            f = (mas[i][j]<=mas[i+1][j]) ? 1 : 0;
        if (f)
        {
            for (i = 0; i < rows; i++)
            {
                if(!*(mas2+i))
                    *(mas2+i) = (int*)realloc(*(mas2+i), (g+1)*sizeof(int));
                *(mas2+i)+g = *(mas+i)+j);
            }
            g++;
        }
        *ncol = g;
        return mas2;
    }
}
//*****
//Функция освобождения памяти
void fr(int** mas, int rows)
{
    if (mas!=NULL)
    {
        for (int i = 0; i<rows; i++)
            free(*(mas+i));
        free(mas);
        mas = NULL;
    }
}

```

Результаты решения задачи

При выполнении программы были получены результаты, совпадающие со значениями, приведенными в Таблице 1. Ошибок не обнаружено.

Вывод

При выполнении лабораторной работы были получены практические навыки использования функций и указателей на языке программирования «C/C++».