**Федеральное государственное бюджетное**
**образовательное учреждение высшего образования**
**"Московский авиационный институт**
**(Национальный исследовательский университет)"**

Институт №8 "Информационные технологии и прикладная математика"

Кафедра №806 "Вычислительная математика и программирование"

# К У Р С О В А Я  Р А Б О Т А

**По курсу "Практикум программирования"**

**Задание №7.**

**"Разреженные матрицы"**

Выполнил: студент группы М8О-104Б-22
Тесля Данила Сергеевич
Руководитель:
Потенко Максим Алексеевич
Дата: 04.10.2023
Оценка:.....

Москва - 2023 г.

# Содержание

# 1 Постановка задачи

Составить программу на языке Си с функциями для обработки прямоугольных разреженных матриц.

Вариант № [3, 11]


Транспонировать разреженную матрицу относительно побочной диагонали и выяснить, является ли эта матрица кососимметрической.

# 2 Ход решения

## 2.1 Алгоритм

Перед выполнением работы необходимо изучить материалы по созданию и обработке разреженных матриц на языке Си, а также изучить основы модульного программирование на Си и автоматизацию сборки программ модульной структуры на Си с использованием утилиты make. Создадим программу из трех файлов, один из которых будет главным файлом программы, в нем будет функция main, во втором файле будет реализация всех необходимых функций программы, и третий – заголовочный файл с определениями структур и функций. Создадим тестовые файлы с заготовленными данными – с которыми будет работать программа. Так же дополнительно создадим файл в котором будут команды – указания для корректной сборки и компиляции программы, запускаться данный файл будет с помощью утилиты make. В заголовочном файле (.h) определим структуру вектора – далее в программе 3 таких вектора будут содержать значения матрицы, и структуру самой матрицы, состоящей из трех векторов – первых из которых содержит индекс строки, в которой находится элемент, второй – индекс столбца, в котором содержится элемент, и третий – содержит значения элемента с данными индексами. Далее в файле для реализации функций необходимо будет реализовать функции создания вектора, создания матрицы, добавления элемента в конец вектора, печати вектора, печати матрицы в разреженном виде, печать матрицы в стандартном виде, транспонирования относительно побочной диагонали, проверка

матрицы на кососимметричность, удаления вектора, удаления матрицы. Когда все эти функции будут реализованы – перейдем главному файлу программы – в функции main, в нем необходимо будет считать в структуру матрицы – матрицу из тестового файла, а затем сделаем контекстное меню, где пользователь сможет произвести действия с матрицей по выбору – печать матрицы в разреженном виде, печать матрицы в стандартном виде, транспонирование матрицы относительно побочной диагонали, проверка матрицы на кососимметричность, сделать это можно например конструкцией switch case в цикле while, и считывать выбор пользователя, пока он не захочет завершить работу с матрицей.

## 2.2 Решение

Создадим текстовый файл Makefile, содержащий набор инструкций о том, как компилировать и линковать исходный код программы. Создадим тестовые файлы с заготовленными матрицами – с которыми будет работать программа – input.txt, input1.txt input2.txt. Создадим программу из трех файлов, один из которых будет главным файлом программы – kp9main.c, в нем будет функция main, во втором файле matrix.c будет реализация всех необходимых функций программы, и третий – заголовочный файл matrix.h с определениями структур и функций. В заголовочном файле для начала воспользуемся директивами #ifndef –MATRIX-H #define -MATRIX-H #endif методом условной компиляции, для предотвращения повторной компиляции кода, подключим стандартные библиотеки языка Си. Далее в этом же файле введем определение структуры вектора - vector и структуры разреженной матрицы matrix, у структуры вектора поля будут int* elem – массив значений элементов вектора int size – размер вектора, у структуры матрицы полями будут vector* row; - вектор хранящий значения индексов строк каждого из элементов, vector* col; вектор хранящий значения индексов столбцов каждого из элементов, vector* value; вектор хранящий значения элементов матрицы. Также в заголовочном файле определим конструкцию enum для ее дальнейшего использования в операторе switch в основном файле программы. Далее в файле для реализации функций необходимо будет реализовать функции. Функции создания вектора vector-create – в функции выделяется память под структуру вектора, поля вектора зануляются – создан пустой вектор, возвращен указатель на этот вектор, создание матрицы –matrix-create - выделяется память под структуру матрицы, трем полям струкутры матрицы присваиваются указатели на векторы, созданные путем вызова функции создания вектора для mat->row, mat->col и mat->value. Добавления элемента в конец вектора vector-pushback – в функции происходит увеличение размера вектора на 1 и увеличение блока выделенной памяти – и добавления элемента в конец вектора. Печати век-

тора - vector-print – происходит проход по элементам вектора и их вывод на экран, печати матрицы в разреженном виде - spmatrix-print – трижды вызывается функция vector-print для каждого из веторов, задающих матрицу, печать матрицы в стандартном виде - stdmatrix-print – функция принимает указатель на структуру разреженной матрицы, и размеры матрицы, считанные из тестового файла в функции main – вводится индексация для обхода матрицы в обычном представлении – по индексам i j – и производится проверка, есть ли в разреженной матрице элементы с такими индексами, если нет, то элемент= 0 и в разреженную матрицу он записан не был, таким образом обходится вся матрица размеров m на n, транспонирования относительно побочной диагонали – transpose – в функцию передается указатель на разреженную матрицу и размеры стандартной матрицы, возвращается указатель на транспонированную относительно побочной диагонали матрицу, в теле функции происходит транспонирование матрицы относительно побочной диагонали по правилу такого траспонирования в математике – используются различные проходы по векторам разреженной матрицы с помощью итераторов, создается новая структура для записи в нее транспонированной матрицы – и в определенном порядке с помощью функции добавления элемента в конец вектора – производится заполнение векторов, которые определяют новую разреженную – транспонированную матрицу, случаи будут – для квадратной матрицы, для той, где строк больше чем столбцов, и для той, где столбцов больше чем строк – в каждом случае правило транспонирования относительно побочной диагонали будет отличаться – следовательно и порядок записи будет различным. Проверка матрицы на кососимметричность - is-symmetric – функция возвращает true если матрица кососимметрическая, и false если нет, в теле функции происходит проверка условий кососимметричности матрицы – путем прохода по векторам разреженной матрицы и сравнению их в определенном порядке друг с другом, удаления вектора- vector-free – очистка памяти вектора, зануление полей структуры вектора, удаления матрицы- matrix-free - очистка памяти матрицы, зануление полей структуры матрицы – путем вызова функции удаления вектора для каждого из полей.

Перейдем к главному файлу программы – в функции main, в нем необходимо считать в структуру матрицы – матрицу из тестового файла – создается матрицы spmatrix путем вызова соответствующих функций – открывается для чтения тестовый файл – считываются сначала размеры матрицы m и n – затем построчно считываются ненулевые элементы матрицы и записываются в структуру разреженной матрицы,- таким образом разреженная матрица считана в программу. Затем сделаем контекстное меню, где пользователь сможет произвести действия с матрицей по выбору – печать матрицы в разреженном виде, печать матрицы в стандартном виде, транспонирование матрицы относительно побочной диагонали, проверка

матрицы на кососимметричность, сделать это можно например конструкцией switch case в цикле while, и считывать выбор пользователя, пока он не захочет завершить работу с матрицей.

# 3 Код программы и тесты

https://github.com/tesla-2002/kp7

```
tesla-2002@tesla2002-HP-ProBook-430-G7:~/labs/kp7$ valgrind --leak-check=full -s  ./kp7.out ./input.txt
==3001== Memcheck, a memory error detector
==3001== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==3001== Using Valgrind-3.21.0 and LibVEX; rerun with -h for copyright info
==3001== Command: ./kp7.out ./input.txt
==3001==
Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
1
(  0   0   1   1   2   2 )
(  1   2   0   2   0   1 )
(  1  -3  -1  -2   3   2 )

Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
2
  0   1  -3
 -1   0  -2
  3   2   0

Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
3
Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
1
(  0   0   1   1   2   2 )
(  1   2   0   2   0   1 )
( -2  -3   2   1   3  -1 )
```

```
Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
2
  0  -2  -3
  2   0   1
  3  -1   0

Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
4
Да, эта матрица - кососимметрическая
Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
0
==3001==
==3001== HEAP SUMMARY:
==3001==     in use at exit: 0 bytes in 0 blocks
==3001==   total heap usage: 48 allocs, 48 frees, 7,264 bytes allocated
==3001==
==3001== All heap blocks were freed -- no leaks are possible
==3001==
==3001== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
tesla-2002@tesla2002-HP-ProBook-430-G7:~/labs/kp7$
```

```
tesla-2002@tesla2002-HP-ProBook-430-G7:~/labs/kp7$ valgrind --leak-check=full -s  ./kp7.out ./input1.txt
==3056== Memcheck, a memory error detector
==3056== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==3056== Using Valgrind-3.21.0 and LibVEX; rerun with -h for copyright info
==3056== Command: ./kp7.out ./input1.txt
==3056==
Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
1
(  0   1   1   3   4   4 )
(  0   0   1   2   0   1 )
(  1  15   4   2   3   4 )

Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
2
  1   0   0
 15   4   0
  0   0   0
  0   0   2
  3   4   0

Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
3
Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
1
(  0   1   1   2   2   2 )
(  4   1   4   1   2   3 )
(  3   4   4  15   1   2 )
```

```
Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
2
  0   0   0   0   3
  0   4   0   0   4
  0  15   1   2   0

Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
4
Нет, эта матрица не явл. кососимметрической
Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
0
==3056==
==3056== HEAP SUMMARY:
==3056==     in use at exit: 0 bytes in 0 blocks
==3056==   total heap usage: 48 allocs, 48 frees, 7,264 bytes allocated
==3056==
==3056== All heap blocks were freed -- no leaks are possible
==3056==
==3056== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
tesla-2002@tesla2002-HP-ProBook-430-G7:~/labs/kp7$
```

```
tesla-2002@tesla2002-HP-ProBook-430-G7:~/labs/kp7$ valgrind --leak-check=full -s  ./kp7.out ./input2.txt
==3104== Memcheck, a memory error detector
==3104== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==3104== Using Valgrind-3.21.0 and LibVEX; rerun with -h for copyright info
==3104== Command: ./kp7.out ./input2.txt
==3104==
Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
1
(  0   1   1   1   1   2   2   2 )
(  5   0   1   3   4   0   2   5 )
(  1   2   3   5   4   7   6  -3 )

Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
2
  0   0   0   0   0   1
  2   3   0   5   4   0
  7   0   6   0   0  -3

Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
3
Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
1
(  0   1   2   2   3   4   5   5 )
(  0   1   0   1   1   1   0   2 )
(  6   3   7   2   5   4   1  -3 )
```

```
Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
2
  6   0   0
  0   3   0
  7   2   0
  0   5   0
  0   4   0
  1   0  -3

Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
4
Нет, эта матрица не явл. кососимметрической
Что Вы хотите сделать?
 [1]-Вывести разреженную матрицу
 [2]-Вывести матрицу в привычном виде
 [3]-Транспонировать матрицу отн. побочн. диагонали
 [4]-Проверить, является ли она кососимметрической
 [0]-Завершить
0
==3104==
==3104== HEAP SUMMARY:
==3104==     in use at exit: 0 bytes in 0 blocks
==3104==   total heap usage: 60 allocs, 60 frees, 7,624 bytes allocated
==3104==
==3104== All heap blocks were freed -- no leaks are possible
==3104==
==3104== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
tesla-2002@tesla2002-HP-ProBook-430-G7:~/labs/kp7$
```

# 4 Описание программы

## 4.1 Описание функций

| | |
|---|---|
| vector-create | функции создания вектора |
| matrix-create | функция создания матрицы |
| vector-pushback | функция добавления элемента в конец вектора |
| spmatrix-print | печать матрицы в разреженном виде |
| stdmatrix-print | печать матрицы в стандартном виде |
| transpose | транспонирование матрицы отн. побочной диагонали |
| is-symmetric | проверка матрицы на кососимметричность |
| vector-free | удаление вектора |
| matrix-free | удаление матрицы |
| main | главная функция программы |

## 4.2 Описание ПО

| | |
|---|---|
| ПК | HP ProBook 430 G7 |
| ОС | Ubuntu 22.04.3 LTS |
| Интерпретатор команд | Bash |
| Редактор текстов | Sublime |
| Язык программирования | C |
| Местонахождение файла | /home/labs |

# 5 Заключение

В ходе выполнения данной лабораторной работы я изучил и научился работать на практике с разреженными матрицами на языке Си. Получил навыки считывания матрицы из файла и составления по ней разреженной матрицы в программе, научился основным методам обработки разреженных матриц – добавления элемента, печати матрицы в разреженном и стандартном виде, а так же выполнять более сложные задания – такие как транспонирование разреженных матриц и проверки свойств таких матриц. Все полученные мной знания и навыки пригодятся в выполнении лабораторных и курсовых работ в будущем.