

Вопросы для самостоятельного изучения к заданию VII курсового проекта

РАЗРЕЖЕННЫЕ МАТРИЦЫ

- 1. Представление массивов в памяти ЭВМ.
- 2. Адресация элементов массивов и ее использование для представления структур данных.
- 3. Передача параметров-массивов и параметров-записей.
- 4. Ошибки адресации массивов и их последствия при выполнении программ в операционных системах с защитой памяти и без неё.
- 5. Приемы обработки и ввода/вывода массивов на скалярных ЭВМ.
- 6. Представление обычных и вариантных записей (структур) в памяти ЭВМ.
- 7. Приемы обработки и ввода/вывода записей.
- 8. Разреженные матрицы. Их представление в памяти ЭВМ.
- 9. Особенности хранения в памяти ЭВМ треугольных, симметричных и квазидиагональных матриц.
- 10. Приемы хранения и обработки разреженных матриц на языке Си.

Задание VII. Разреженные матрицы.

Составить программу на языке Си с процедурами и/или функциями для обработки *прямоугольных* разреженных матриц с элементами целого (группы 6, 8), вещественного (группы 2-5), или комплексного (группы 1, 7) типов, которая:

- 1. вводит матрицы различного размера, представленные во входном текстовом файле в обычном формате (по строкам), с одновременным размещением ненулевых элементов в разреженной матрице в соответствии с заданной схемой;
- 2. печатает введенные матрицы во внутреннем представлении согласно заданной схеме размещения и в обычном (естественном) виде;
- 3. выполняет необходимые преобразования разреженных матриц (или вычисления над ними) путем обращения к соответствующим процедурам и/или функциям;
- 4. печатает результат преобразования (вычисления) согласно заданной схеме размещения и в обычном виде.

В процедурах и функциях предусмотреть проверки и печать сообщений в случаях ошибок в задании параметров. Для отладки использовать матрицы, содержащие 5–10% ненулевых элементов с максимальным числом элементов 100.

Вариант схемы размещения матрицы определяется по формуле  $((N + 3) \% 4) + 1$ , где  $N$  — номер студента по списку в группе. Вариант преобразования определяется по формуле  $((N - 1) \% 11) + 1$ . Вариант физического представления (1 — отображение на массив, 2 — отображение на динамические структуры) определяются по формуле  $([1.5 \times ((3 + M) \% 9)] + N) \% 2 + 1$ , где  $M$  — номер группы. В случае использования динамических структур индексы заменяются соответствующими ссылками.

**Варианты схемы размещения матрицы:** все матрицы  $m \times n$  хранятся *по строкам*, в порядке возрастания индексов ненулевых элементов.

1. Цепочка ненулевых элементов в векторе A со строчным индексированием (индексы в массиве M равны 0, если соответствующая строка матрицы содержит только нули)

M:	Индекс начала 1-ой строки в массиве A		Индекс начала 2-й строки	...	Индекс начала N-ой Строки	
A:	Номер столбца	Значение	Индекс следующего ненулевого элемента этой строки (или 0)	Номер столбца	Значение	Индекс следующего ненулевого элемента этой строки (или 0)
						...

Индекс, равный нулю, означает отсутствие ненулевых элементов в строке (или в ее остатке). Если матрицы не изменяются программой, возможна экономия памяти за счет отказа от хранения в массиве A индексов следующего элемента столбца (когда элементы идут подряд). Вставка и удаление при этом способе возможны, но чересчур дороги: число перестановок элементов составит  $O(N)$  вместо  $O(1)$ .

2. Один вектор:

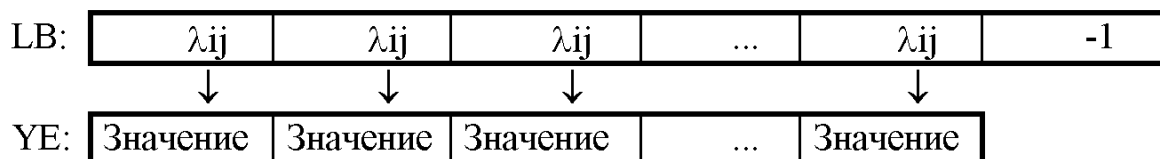
Ненулевому элементу соответствуют две ячейки: первая содержит номер столбца, вторая содержит значение элемента. Нуль в первой ячейке означает конец строки, а вторая ячейка содержит в этом случае номер следующей хранимой строки. Нули в обеих ячейках являются признаком конца перечня ненулевых элементов разреженной матрицы.

0	Номер строки	Номер столбца	Значение	Номер столбца	Значение	...
...						
0	Номер строки	Номер столбца	Значение	...	0	0

### 3. Три вектора:



### 4. Два вектора:



где  $\lambda_{ij} = (i - 1) \times n + j - 1$

По согласованию с преподавателем возможна модификация схемы хранения, например хранение не исходной, а транспонированной матрицы.

*Дополнительное задание:* реализовать функциональную спецификацию АТД Матрица.

### Варианты преобразований:

1. Определить максимальный по модулю элемент матрицы и разделить на него все элементы строки, в которой он находится. Если таких элементов несколько, обработать каждую строку, содержащую такой элемент.
2. Определить максимальный по модулю элемент матрицы и разделить на него все элементы столбца, в котором он находится. Если таких элементов несколько, обработать предпоследний столбец, содержащий такой элемент.
3. Найти элемент матрицы, ближайший к заданному значению. Разделить на него элементы строки и столбца, на пересечении которых он расположен. Если таких элементов несколько, обработать все.
4. Умножить разреженную матрицу на вектор-столбец и вычислить количество ненулевых элементов результата.
5. Умножить вектор-строку на разреженную матрицу и вычислить количество ненулевых элементов результата.
6. Вычислить сумму двух разреженных матриц. Проверить, не является ли полученная матрица симметричной.
7. Найти строку, содержащую наибольшее количество ненулевых элементов, и напечатать ее номер и сумму элементов этой строки. Если таких строк несколько, обработать все.
8. Вычислить произведение двух разреженных матриц. Проверить, не является ли полученная матрица диагональной.
9. Найти столбец, содержащий наибольшее количество ненулевых элементов, и напечатать его номер и произведение элементов этого столбца. Если таких столбцов несколько обработать предпоследний.
10. Вычислить матричный многочлен первой степени от разреженной матрицы:  $(a \cdot M + b \cdot E)$ , где  $E$  — единичная матрица,  $a$  и  $b$  — числовые константы.
11. Транспонировать разреженную матрицу относительно побочной диагонали. Выяснить, является ли полученная матрица кососимметрической.

### Литература к заданию VII и к лабораторной работе № 21

1. Никулин С.П. Представление и обработка данных с регулярной структурой // Под. ред. Зайцева В.Е. — М.: МАИ, 1997.
2. Берзтисс А.Т. Структуры данных. — М.: Мир, 1974.
3. Тьюарсон Р. Разреженные матрицы. — М.: Мир, 1977.
4. Писсанецки С. Технология разреженных матриц. — М.: Мир, 1980.
5. Карасев С.Б., Кошелева Т.Я., Чернышов Л.Н. Машинные алгоритмы обработки информации. — М.: Изд-во МАИ, 1987.
6. Кнут Д. Искусство программирования для ЭВМ. Т 1. Основные алгоритмы. — М.: Мир, 1976.
7. Кристиан К. Введение в операционную систему UNIX. — М.: Финансы и статистика, 1985.
8. Беляков И.Н., Рабовер Ю.И., Фридман А.Л. Мобильная операционная система: Справочник. — М.: Радио и связь, 1991.
9. Баурн С. Операционная система UNIX. — М.: Мир, 1986.
10. Тихомиров В.П., Давидов М.И. Операционная система ДЕМОС: инструментальные средства программирования. — М.: Статистика, 1988. (C Shell, make, lex, yacc).
11. Тейнсли Д. Linux и UNIX: программирование в shell. Руководство разработчика: Пер. с англ. — К.: Издательская группа BHV, 2001. — 464 с.
12. Вирт Н. От Модуля к Оберону // Системная информатика. Вып.1. — Новосибирск: Наука, 1991, с.63–75.
13. Вирт Н. От разработки языка программирования к созданию компьютера. // Лекции лауреатов премии Тьюринга. — М.: Мир, 1993, с.210–223
14. FTP-архив Интернет МТИ: <ftp://prep.ai.mit.edu/pub/gnu>

15. Айлифф Дж. Принципы построения базовой машины. -М.: Мир, 1974.
16. Еремин А.Ю., Марьяшкин Н.Я. Пакет программ SPARSE для решения систем линейных алгебраических уравнений с разреженными матрицами. -М.: ВЦ АН СССР, 1978.
17. Еремин А.Ю., Марьяшкин Н.Я. Пакет программ SOLVER системы нелинейных функциональных и обыкновенных дифференциальных уравнений с разреженными якобиевыми матрицами. -М.: ВЦ АН СССР, 1980.
18. Белаи В.О., Марьяшкин Н.Я. Пакет программ СЛАУРМ. Решение систем с разреженными матрицами. -М.: ВЦ АН СССР, 1989.

## Вопросы для самостоятельного изучения к заданию VIII курсового проекта

### ISO/IEC 9899:1999 Programming languages — C [C99]

1. Краткая историческая справка о предшественниках C: A, B, BCPL.
2. Стандарты языка C: C89 и C99. Новые возможности. Совместимость.
3. Дальнейшее развитие языка C: C++. Новые возможности. Совместимость.
4. Дополнительные типы данных C99: комплексный тип, целые типы. Булевский тип, заголовок <stdbool.h>.
5. Константные выражения. Недесятичная запись целых констант. Заголовочные файл <limits.h>, <stdint.h>.
6. Операторы сдвига << и >>. Операторы @=. Операторы \*, & и ->. Оператор sizeof.
7. Строковый тип. Библиотека <string.h>.
8. Математическая библиотека <math.h>.
9. Функции в качестве типов и аргументов. Оператор typedef.
10. Структуры. Инициализация структур в C89 и C99.
11. Работа с файлами: библиотека <stdio.h>. Тип FILE, функции fopen, fclose, fread, fwrite, fscanf, fprintf, fseek. Стандартные файлы.
12. Выбор стандарта C. Опция -std.
13. Стандарт языка об особенностях вычисления логических выражений. Примеры.

### СПИСКОВЫЕ СТРУКТУРЫ В ЯЗЫКАХ ПРОГРАММИРОВАНИЯ.

1. Моделирование списков на линейной памяти с прямым и последовательным доступом. Итераторы.
2. Списки в динамической памяти Си.
3. Библиотечные средства обработки списков в C++.
4. \*Библиотечные средства обработки списков в Java.
5. \*Представление списков деревьями (Пролог).
6. \*Понятие о существенно списковых языках программирования (Лисп).
7. \*Встроенные списки языка Python. Особенности индексации. Срезы. Встроенные операции над списками.

## Задание VIII. Линейные списки

Составить и отладить программу на языке Си для обработки линейного списка заданной организации с отображением списка на динамические структуры (группы 1, 2, 3, 8) или на массив (только с индексным доступом, без применения ссылок и указателей, для групп 4, 5, 6, 7). Навигацию по списку следует реализовать с применением итераторов. Предусмотреть выполнение одного нестандартного и четырех стандартных действий:

1. Печать списка.
2. Вставка нового элемента в список.
3. Удаление элемента из списка.
4. Подсчет длины списка.

**ТИП ЭЛЕМЕНТА СПИСКА:** (определяется как *номер\_группы* % 8 + 1):

1. Целый.
2. Вещественный.
3. Перечислимый.
4. Строковый.
5. Литерный.
6. Комплексный.
7. Машинное слово.
8. Булевский
9. \*BCD
10. \*Множество.
11. \*Ссылочный.
12. \*Процедурный.