1. Цель работы:

изучение псевдодинамического распределения памяти на языке ФОРТРАН; изучение форматов хранения матриц большой размерности; оптимизация программ по точности, скорости, памяти; изучение погрешности вычисления скалярного произведения; изучение способов отладки; изучение принципов формирования тестов для вычислительных программ; изучение файлов прямого доступа.

2. Задание:

1) Написать программу, реализующую на языке ФОРТРАН требуемые действия над матрицами произвольной (задаваемой пользователем) размерности с учетом следующих требований:

- для распределения памяти под необходимые массивы использовать идеи псевдодинамической памяти: память выделять в головной программе, описав 1 одномерный массив максимально возможной для данного компьютера размерности; вся оставшаяся после распределения память должна находиться в конце этого массива; при недостатке памяти выдавать соответствующее сообщение;
- для вариантов с ленточным и диагональными форматами в подпрограммах матрицы хранить в виде двумерного массива;
- каждое действие (ввод данных, вывод результата, перемножение и т.д.) должно быть реализовано в виде подпрограммы; для тех вариантов, в которых возможно реализовать перемножение строки на столбец с помощью перемножения одномерных векторов, обязательно наличие такой подпрограммы;
- вычисления должны производиться оптимальным образом;
- все входные данные должны вводиться из файлов (матрицы и векторы в разных файлах); размерности всех объектов (а также любая вспомогательная скалярная информация, например, ширина ленты) хранятся в отдельном текстовом файле, например, два целых числа для прямоугольной плотной матрицы, одно – для вектора); при этом матрица в файлах хранится уже в заданном формате, число файлов для хранения матрицы определяется форматом и обычно совпадает с числом массивов.

2) Протестировать разработанную программу. Для тестирования использовать матрицы небольшой размерности.

3) Реализовать задание лабораторной работы с использованием написанных ранее подпрограмм при условии, что матрицы и векторы хранятся в файлах прямого доступа (файл с размерностями оставить текстовым).

4) Разработать программу генерации тестов большой размерности в заданном формате. Программа должна создавать все требуемые файлы прямого доступа.

5) Для просмотра и создания тестов с файлами прямого доступа рекомендуется написать дополнительные программы, которые формируют файл прямого доступа по заданному текстовому файлу, и наоборот.

6) Для просмотра матриц малой размерности, заданных в разреженном и профильном формате, рекомендуется разработать программу, выводящую матрицу в плотном формате в текстовый файл.
Вариант: Умножение несимметричной матрицы на вектор. Матрица в памяти хранится в профильном формате: *столбцово-строчный формат* (нижний треугольник хранится по столбцам, а верхний – по строкам);

3. Анализ задачи:

Для хранения квадратной матрицы размерности n в профильном формате используется следующая структура данных:

- Вещественный массив **di** . Этот массив имеет размерность n и содержит последовательно диагональные элементы матрицы.
- Вещественные массивы **al** и **au** для хранения внедиагональных элементов матрицы нижнего (по столбцам) и верхнего (по строкам) треугольников матрицы соответственно.
- Целочисленный массив **ia** для хранения информации о профиле. Элемент ia (k) равен индексу (в нумерации 1), с которого начинаются элементы k -го столбца (строки) в массивах al и au . Размерность массива ia равна n +1, причем ia (n +1) равен индексу первого незанятого элемента в массивах al и au (т.е. размерность этих массивов равна ia (n +1) -1). Разность ia (i +1) - ia (i) равна значению профиля i -й столбца (строки) нижнего (верхнего) треугольника. Очевидно, что ia(n)=ia(n+1)

Пример, профильная матрица 10×10

$$A := \begin{pmatrix} a_{11} & a_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{12} & a_{22} & a_{23} & a_{24} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & 0 & 0 & 0 & 0 \\ 0 & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & 0 & 0 \\ 0 & 0 & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} & a_{510} \\ 0 & 0 & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} & a_{610} \\ 0 & 0 & 0 & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} & a_{710} \\ 0 & 0 & 0 & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} & a_{89} & a_{810} \\ 0 & 0 & 0 & 0 & a_{95} & a_{96} & a_{97} & a_{98} & a_{99} & a_{910} \\ 0 & 0 & 0 & 0 & a_{105} & a_{106} & a_{107} & a_{108} & a_{109} & a_{1010} \end{pmatrix}$$

$di = \{a_{11}, a_{22}, a_{33}, a_{44}, a_{55}, a_{66}, a_{77}, a_{88}, a_{99}, a_{1010}\}$

$ia = \{1,2,4,7,11,16,20,23,25,26,26\}$

$al = \{a_{21}, a_{32}, a_{42}, a_{43}, a_{53}, a_{63}, a_{54}, a_{64}, a_{74}, a_{84}, a_{65}, a_{75}, a_{85}, a_{95}, a_{105},$

$a_{76}, a_{86}, a_{96}, a_{106}, a_{87}, a_{97}, a_{107}, a_{98}, a_{108}, a_{109}\}$

$au = \{a_{12}, a_{23}, a_{24}, a_{34}, a_{35}, a_{36}, a_{45}, a_{46}, a_{47}, a_{48}, a_{56}, a_{57}, a_{58}, a_{59}, a_{510},$

$a_{67}, a_{68}, a_{69}, a_{610}, a_{78}, a_{79}, a_{710}, a_{89}, a_{810}, a_{910}\}$

4. Решение:

Доступ к элементам:

$a[i,j] = al[ia[j] + i - j - 1]$, если $i > j$

$a[i,j] = au[ia[i] + j - i - 1]$, если $i < j$

$a[i,j] = di[i]$, если $i = j$

Умножение профильной матрицы на вектор:

$DO\ i = 1, n$

$res(i) = res(i) + di(i)$

$DO\ j = 1, ia(i+1) - ia(i)$

$res(j+i) = res(j+i) + al(ia(i) + j - 1) * vec(i)$

$res(i) = res(i) + au(ia(i) + j - 1) * vec(i+j)$

$END\ DO$

$END\ DO$

Входные данные:

Текстовый файл «size.txt», содержащий информацию о размере матрицы, и файлы прямого доступа «matrix.bin» и «vector.bin»

Выходные данные:

Текстовый файл «result.txt» с результатом умножения

Псевдодинамическая память:

В головной программе выделяется память под 1 одномерный массив размером nmax; все данные программы хранятся в нем на следующих позициях (n – размер матрицы , k=a(2n+1) – размер au и al +1):

$di: a(1) - a(n)$

$ia: a(n+1) - a(2n+1)$

$al: a(2n+2) - a(2n+k)$

$au: a(2n+k+1) - a(2n+2k-1)$

$vector: a(2n+2k) - a(3n+2k-1)$

$result: a(3n+2k) - a(4n+2k-1)$

Генерация тестов:

n – размер матрицы. Профиль матрицы с каждой строкой (столбца) матрицы увеличивается на 1, пока не достигнет своего максимального значения – n/2; профиль столбцов нижнего треугольника заполняется 1, профиль сток верхнего треугольника – 2, диагональ – 3.

Размер al и au: пусть k=n , если n-четное, и k=n-1 , если n-нечетное, т.е.

$k = n - (n\ mod\ 2)$, тогда $2 * \left(\frac{1+\left(\frac{k}{2}-1\right)}{2} * \left(\frac{k}{2} - 1\right) \right) + \frac{k}{2} = \frac{k^2}{4}$

5. Описание подпрограмм:

Название	Формальные параметры	Назначение
(SUB) error	i - номер ошибки	обработка ошибок
(SUB) convert_if	n – размер матрицы di,ia,al,au	преобразование профильный матрицы в плотный формат и запись в файл
(FUN) i_size	n – размер матрицы ia – общая матрица	Возвращает значение ia(n+1) - размер матриц al и au
(SUB) float_data_iof	mode – тип: 1 – чтение текстового файла, 2 – чтение бинарного файла, 3 – запись бинарного файла, 4 – запись текстового файла; a – общая матрица; i1,i2 – позиции элементов в общей матрице; j – нач. позиция записи в файле; file_name	запись/чтение данных вещественного формата;
(SUB) int_data_iof	mode,ia,i1,i2,j,file_name	запись/чтение данных целого формата (аналогично float_data_iof)
(SUB) matrix_io	mode,a,n,mtrx_file	запись/чтение матрицы
(SUB) text_bin_files	a, mode – тип: 1 – текстовый в двоичный, 2 – двоичный в текстовый;	
(SUB) vector_io	mode,a,n,vec_file	запись/чтение вектора;
(SUB) result_o	a,n	запись результата
(SUB) matrix_generate	n,di,ia,al,au	генерация матрицы заданной размерности
(SUB) vector_generate	n,vec	генерация вектора заданной размерности
(SUB) data_generate	n,a	проверка на переполнение и вызов подпрограмм matrix_generate и vector_generate
(SUB) multiplication	n,di,ia,al,au,vec,res	умножения матрицы на вектор

6. Тесты:

Вид программы:

```
1 - read matrix out of file
2 - read vector out of file
3 - generate data
4 - convert to plotniy format
5 - convert text or bin files
6 - matrix multiplication
7 - exit
```

Генерация данных размером 5 (матрица 5 x 5, вектор 5 x 1), преобразование двоичный файлов в текстовые, преобразование матрицы в плотный формат и нахождение произведения:

di	ia	al	au	Плотный формат	Вектор	Рез-т	Ожид
3.0 3.0 3.0 3.0 3.0	1 2 4 5 5 5	1.0 1.0 1.0 1.0	2.0 2.0 2.0 2.0	3.0 2.0 0.0 0.0 0.0 1.0 3.0 2.0 2.0 0.0 0.0 1.0 3.0 2.0 0.0 0.0 1.0 1.0 3.0 0.0 0.0 0.0 0.0 0.0 3.0	1.0 1.0 1.0 1.0 1.0	5.0 8.0 6.0 5.0 3.0	$\begin{pmatrix}5\\8\\6\\5\\3\end{pmatrix}$