

Требования к программам

1. Программа должна получать все параметры в качестве аргументов командной строки.
2. Во всех задачах требуется вычислить вектор, являющийся m -м элементом указанной в условии последовательности, строящейся по заданной $n \times n$ матрице A и 0-му элементу последовательности x_0 .
3. Аргументы командной строки для задач 1, 3–6:
 - 1) m – номер элемента последовательности,
 - 2) n – число строк и столбцов $n \times n$ матрицы A ,
 - 3) p – количество выводимых значений в матрице и векторах,
 - 4) k_A – задает номер формулы для инициализации матрицы A , должен быть равен 0 при вводе матрицы A из файла,
 - 5) f_A – имя файла, откуда надо прочитать матрицу A . Этот аргумент **отсутствует**, если $k_A \neq 0$,
 - 6) k_x – задает номер формулы для инициализации вектора x_0 размера $n \times 1$, должен быть равен 0 при вводе вектора x_0 из файла,
 - 7) f_x – имя файла, откуда надо прочитать вектор x_0 . Этот аргумент **отсутствует**, если $k_x \neq 0$.

Например, запуск

```
./a.out 100 6 5 0 a.txt 0 x.txt
```

означает, что матрицу A 6×6 надо прочитать из файла `a.txt`, вектор x_0 длины 6 надо прочитать из файла `x.txt`, выводить не более 5 строк и столбцов матрицы и вектора, вычислить требуется 100-й элемент последовательности x_{100} ; а запуск

```
./a.out 100 2000 6 1 4
```

означает, что матрицу A 2000×2000 надо инициализировать по формуле номер 1, вектор x_0 длины 2000 надо инициализировать по формуле номер 4 (как матрицу 2000×1), и выводить не более 6-ти строк и столбцов матрицы и вектора, вычислить требуется 100-й элемент последовательности x_{100} .

4. Аргументы командной строки для задач 2, 7–10:
 - 1) τ – параметр τ подпрограммы,
 - 2) m – номер элемента последовательности,
 - 3) n – число строк и столбцов $n \times n$ матрицы A ,
 - 4) p – количество выводимых значений в матрице и векторах,
 - 5) k_A – задает номер формулы для инициализации матрицы A , должен быть равен 0 при вводе матрицы A из файла,
 - 6) f_A – имя файла, откуда надо прочитать матрицу A . Этот аргумент **отсутствует**, если $k_A \neq 0$,

- 7) k_x – задает номер формулы для инициализации вектора x_0 размера $n \times 1$, должен быть равен 0 при вводе вектора x_0 из файла,
- 8) f_x – имя файла, откуда надо прочитать вектор x_0 . Этот аргумент **отсутствует**, если $k_x \neq 0$.

5. Ввод матрицы должен быть оформлен в виде подпрограммы, находящейся в отдельном файле.
6. Ввод матрицы из файла. В указанном файле находится матрица в формате:

$$\begin{array}{ccc} a_{1,1} & \dots & a_{1,n} \\ a_{2,1} & \dots & a_{2,n} \\ \dots & \dots & \dots \\ a_{m,1} & \dots & a_{m,n} \end{array}$$

где $m \times n$ – указанные размеры матрицы, $A = (a_{i,j})$ – матрица. Программа должна выводить сообщение об ошибке, если указанный файл не может быть прочитан, содержит меньшее количество данных или данные неверного формата.

7. Ввод матрицы и правой части по формуле. Элемент $a_{i,j}$ матрицы A размера $m \times n$ полагается равным

$$a_{i,j} = f(k, m, n, i, j), \quad i = 1, \dots, m, \quad j = 1, \dots, n,$$

где $f(k, m, n, i, j)$ – функция, которая возвращает значение (i, j) -го элемента $m \times n$ матрицы по формуле номер k (аргумент командной строки). Функция $f(k, m, n, i, j)$ должна быть оформлена в виде отдельной подпрограммы.

$$f(k, m, n, i, j) = \begin{cases} \max\{n, m\} - \max\{i, j\} + 1 & \text{при } k = 1 \\ \max\{i, j\} & \text{при } k = 2 \\ |i - j| & \text{при } k = 3 \\ \frac{1}{i + j - 1} & \text{при } k = 4 \end{cases}$$

8. **В задачах 2–10**, где участвует вектор b , он строится после инициализации матрицы $A = (a_{i,j})_{i,j=1,\dots,n}$ по формуле:

$$b = (b_i)_{i=1,\dots,n}, \quad b_i = \sum_{k=0}^{(n-1)/2} a_{i,2k+1}$$

Инициализация должна быть оформлена в виде подпрограммы, вызываемой из функции `main`.

9. Решение должно быть оформлено в виде подпрограммы, находящейся в отдельном файле.
10. Программа должна содержать подпрограмму вывода на экран прямоугольной матрицы $m \times n$ матрицы. Эта подпрограмма используется для вывода исходной $m \times n$ матрицы после ее инициализации, а также для вывода на экран результата работы программы. Подпрограмма выводит на экран не более, чем p строк и столбцов $m \times n$ матрицы, где p – параметр этой подпрограммы (аргумент командной строки). Каждая строка матрицы должна печататься на новой строке, каждый элемент матрицы выводится в строке по формату " %10.3e" (один пробел между элементами и экспоненциальный формат %10.3e).
11. Функция, реализующая задачу, **не должна выделять или использовать дополнительную память**.

12. Сложность работы подпрограммы не должна превышать $C(m+1)*n^2$ при $n \rightarrow \infty, m \rightarrow \infty$. Константа $C = 1$ в задачах 1–7, $C = 3/2$ в задачах 8, 9, $C = 2$ в задаче 10. Это означает, что при переходе от x_{k-1} к x_k

- может быть только одно умножение матрицы A на вектор (во всех задачах),
- надо решать систему линейных уравнений с треугольной матрицей методом последовательного исключения неизвестных (в задачах 8–10).

13. Результатами работы задач 1–10 являются 3 элемента:

- Собственно вектор x_m .
- Два вещественных числа r_1 и r_2 , вычисляемых после вызова задачи:
 - Для задачи 1

$$r_1 = \text{возвращаемое значение функции}, \quad r_2 = \sum_{i=1}^n \left| \left(\sum_{j=1}^n a_{ij} x_j \right) - r_1 x_i \right| / \sum_{i=1}^n |r_1 x_i|$$

- Для задач 2–10

$$r_1 = \sum_{i=1}^n \left| \left(\sum_{j=1}^n a_{ij} x_j \right) - b_i \right| / \sum_{i=1}^n |b_i|, \quad r_2 = \sum_{i=1}^n |x_i - (i \bmod 2)| / \sum_{i=1}^n (i \bmod 2)$$

Здесь $(x_i)_{i=1, \dots, n}$ – это компоненты вектора x_m .

Вычисление r_1 и r_2 должно быть оформлено в виде подпрограммы, вызываемой из функции `main`. Эта подпрограмма **не должна выделять или использовать дополнительную память**.

14. Вывод результата работы функции в функции `main` должен производиться по формату:

- Непосредственно вывод вектора x_m . Он выводится вызовом подпрограммы печати матрицы (см. пункт 10) размера $1 \times n$ (т.е. в строку и **по указанному там формату**)
- Отчет о результате и времени работы:

```
printf ("%s : Task = %d Res1 = %e Res2 = %e Elapsed = %.2f\n",
        argv[0], task, r1, r2, t);
```

где

- `argv[0]` – первый аргумент командной строки (имя образа программы),
- `task` – номер задачи (1–10),
- `r1 = r1` – вычисленное значение r_1 (см. пункт 13),
- `r2 = r2` – вычисленное значение r_2 (см. пункт 13),
- `t` – время работы функции, реализующей решение этой задачи.

Вывод должен производиться в точности в таком формате, чтобы можно было автоматизировать обработку запуска многих тестов.

Задачи

1. Написать функцию, получающую в качестве аргументов $n \times n$ матрицу A , вектора x_0 , x , целые числа n и m , и возвращающую m -й член последовательности $\{\lambda_k\}$, где $\lambda_k = (Ax_k, x_k)/(x_k, x_k)$, $x_k = Ax_{k-1}$, x_0 – 0-й элемент последовательности, значение которого может меняться подпрограммой, (\cdot, \cdot) – евклидово скалярное произведение. В векторе x возвращается значение x_m .
2. Написать подпрограмму, получающую в качестве аргументов $n \times n$ матрицу A , вектора x_0 , b , x , целые числа n , m и вещественное число τ , и возвращающую в векторе x m -й член последовательности $\{x_k\}$, где $(x_k - x_{k-1})/\tau + Ax_{k-1} = b$. x_0 – 0-й элемент последовательности, значение которого может меняться подпрограммой.
3. Написать подпрограмму, получающую в качестве аргументов $n \times n$ матрицу A , вектора x_0 , b , x , r целые числа n и m , и возвращающую в векторе x m -й член последовательности $\{x_k\}$, где $(x_k - x_{k-1})/\tau_{k-1} + Ax_{k-1} = b$, $\tau_k = (r_k, r_k)/(Ar_k, r_k)$, $r_k = Ax_k - b$, (\cdot, \cdot) – евклидово скалярное произведение. x_0 – 0-й элемент последовательности, значение которого может меняться подпрограммой, r – дополнительная память.
4. Написать подпрограмму, получающую в качестве аргументов $n \times n$ матрицу A , вектора x_0 , b , x , r , целые числа n и m , и возвращающую в векторе x m -й член последовательности $\{x_k\}$, где $(x_k - x_{k-1})/\tau_{k-1} + Ax_{k-1} = b$, $\tau_k = (Ar_k, r_k)/(Ar_k, Ar_k)$, $r_k = Ax_k - b$, (\cdot, \cdot) – евклидово скалярное произведение. x_0 – 0-й элемент последовательности, значение которого может меняться подпрограммой, r – дополнительная память.
5. Написать подпрограмму, получающую в качестве аргументов $n \times n$ матрицу A , вектора x_0 , b , x , r целые числа n и m , и возвращающую в векторе x m -й член последовательности $\{x_k\}$, где $D(x_k - x_{k-1})/\tau_{k-1} + Ax_{k-1} = b$, $\tau_k = (D^{-1}r_k, r_k)/(AD^{-1}r_k, D^{-1}r_k)$, $r_k = Ax_k - b$, D – диагональ матрицы A , (\cdot, \cdot) – евклидово скалярное произведение. x_0 – 0-й элемент последовательности, значение которого может меняться подпрограммой, r – дополнительная память.
6. Написать подпрограмму, получающую в качестве аргументов $n \times n$ матрицу A , вектора x_0 , b , x , r , целые числа n и m , и возвращающую в векторе x m -й член последовательности $\{x_k\}$, где $D(x_k - x_{k-1})/\tau_{k-1} + Ax_{k-1} = b$, $\tau_k = (AD^{-1}r_k, r_k)/(AD^{-1}r_k, AD^{-1}r_k)$, $r_k = Ax_k - b$, D – диагональ матрицы A , (\cdot, \cdot) – евклидово скалярное произведение. x_0 – 0-й элемент последовательности, значение которого может меняться подпрограммой, r – дополнительная память.
7. Написать подпрограмму, получающую в качестве аргументов $n \times n$ матрицу A , вектора x_0 , b , x , r , целые числа n , m и вещественное число τ , и возвращающую в векторе x m -й член последовательности $\{x_k\}$, где $D(x_k - x_{k-1})/\tau + Ax_{k-1} = b$, D – диагональ матрицы A . x_0 – 0-й элемент последовательности, значение которого может меняться подпрограммой, r – дополнительная память.
8. Написать подпрограмму, получающую в качестве аргументов $n \times n$ матрицу A , вектора x_0 , b , x , r , w , целые числа n , m и вещественное число τ , и возвращающую в векторе x m -й член последовательности $\{x_k\}$, где $(D + L)(x_k - x_{k-1})/\tau + Ax_{k-1} = b$, D – диагональ матрицы A , L – нижняя треугольная часть матрицы A . x_0 – 0-й элемент последовательности, значение которого может меняться подпрограммой, r , w – дополнительная память.
9. Написать подпрограмму, получающую в качестве аргументов $n \times n$ матрицу A , вектора x_0 , b , x , r , w , целые числа n , m и вещественное число τ , и возвращающую в векторе x m -й член последовательности $\{x_k\}$, где $(D + R)(x_k - x_{k-1})/\tau + Ax_{k-1} = b$, D – диагональ матрицы A ,

R — верхняя треугольная часть матрицы матрицы A . x_0 — 0-й элемент последовательности, значение которого может меняться подпрограммой, r, w — дополнительная память.

10. Написать подпрограмму, получающую в качестве аргументов $n \times n$ матрицу A , вектора x_0, b, x , r, w , целые числа n, m и вещественное число τ , и возвращающую в векторе x m -й член последовательности $\{x_k\}$, где $(D + L)D^{-1}(D + R)(x_k - x_{k-1})/\tau + Ax_{k-1} = b$, D — диагональ матрицы A , L — нижняя треугольная часть матрицы матрицы A , R — верхняя треугольная часть матрицы матрицы A . x_0 — 0-й элемент последовательности, значение которого может меняться подпрограммой, r, w — дополнительная память.