

Требования к программам

1. Задачи оцениваются независимо в двух группах: задачи 1–5 и задачи 6–10.
2. Программа должна получать все параметры в качестве аргументов командной строки.
3. Аргументы командной строки для задач 1–5:
 - 1) n – размерность матрицы,
 - 2) p – количество выводимых значений в матрице,
 - 3) k – задает номер формулы для инициализации матрицы, должен быть равен 0 при вводе матрицы из файла,
 - 4) `filename` – имя файла, откуда надо прочитать матрицу. Этот аргумент **отсутствует**, если $k! = 0$.

Например, запуск

```
./a.out 4 4 0 a.txt
```

означает, что матрицу 4x4 надо прочитать из файла `a.txt` и выводить не более 4-х строк и столбцов матрицы, а запуск

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

означает, что матрицу 2000x2000 надо инициализировать по формуле номер 1 и выводить не более 6-ти строк и столбцов матрицы.

4. Аргументы командной строки для задач 6–7:
 - 1) i – параметр i задачи,
 - 2) j – параметр j задачи,
 - 3) n – число строк матрицы,
 - 4) m – число столбцов матрицы,
 - 5) p – количество выводимых значений в матрице,
 - 6) k – задает номер формулы для инициализации матрицы, должен быть равен 0 при вводе матрицы из файла,
 - 7) `filename` – имя файла, откуда надо прочитать матрицу. Этот аргумент **отсутствует**, если $k! = 0$.

Например, запуск

```
./a.out 1 2 6 4 4 0 a.txt
```

означает, что $i = 1$, $j = 2$, матрицу 6x4 надо прочитать из файла `a.txt` и выводить не более 4-х строк и столбцов матрицы, а запуск

```
./a.out 3 4 2000 4000 6 1
```

означает, что $i = 3$, $j = 4$, матрицу 2000×4000 надо инициализировать по формуле номер 1 и выводить не более 6-ти строк и столбцов матрицы.

5. Аргументы командной строки для задачи 8:

- 1) i – параметр i задачи,
- 2) j – параметр j задачи,
- 3) γ – параметр γ задачи,
- 4) n – число строк матрицы,
- 5) m – число столбцов матрицы,
- 6) p – количество выводимых значений в матрице,
- 7) k – задает номер формулы для инициализации матрицы, должен быть равен 0 при вводе матрицы из файла,
- 8) `filename` – имя файла, откуда надо прочитать матрицу. Этот аргумент **отсутствует**, если $k! = 0$.

6. Аргументы командной строки для задачи 9:

- 1) n – число строк матрицы,
- 2) m – число столбцов матрицы,
- 3) p – количество выводимых значений в матрицах,
- 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 $k_A! = 0$,
- 6) k_B – задает номер формулы для инициализации вектора B размера $m \times 1$, должен быть равен 0 при вводе вектора B из файла,
- 7) f_B – имя файла, откуда надо прочитать вектора B . Этот аргумент **отсутствует**, если $k_B! = 0$.

7. Аргументы командной строки для задачи 10:

- 1) n – число строк матрицы,
- 2) m – число столбцов матрицы,
- 3) k – число столбцов матрицы B ,
- 4) p – количество выводимых значений в матрицах,
- 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 $k_A! = 0$,
- 7) k_B – задает номер формулы для инициализации матрицы B размера $m \times k$, должен быть равен 0 при вводе матрицы B из файла,
- 8) f_B – имя файла, откуда надо прочитать матрицу B . Этот аргумент **отсутствует**, если $k_B! = 0$.

8. Ввод матрицы должен быть оформлен в виде подпрограммы, находящейся в отдельном файле.

9. Ввод матрицы из файла. В указанном файле находится матрица в формате:

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

где $n \times m$ - 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.

10. Ввод матрицы и правой части по формуле. Элемент $a_{i,j}$ матрицы A полагается равным

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

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

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

11. Решение должно быть оформлено в виде подпрограммы, находящейся в отдельном файле.

12. Программа должна содержать подпрограмму вывода на экран прямоугольной матрицы $n \times m$ матрицы. Эта подпрограмма используется для вывода исходной $n \times m$ матрицы после ее инициализации, а также для вывода на экран результата работы программы. Подпрограмма выводит на экран не более, чем p строк и столбцов $n \times m$ 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).

13. Вывод результата задачи в функции `main` должен производиться по формату:

- Непосредственно вывод результата (число или матрица)
- Отчет о времени:

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

где

- `argv[0]` – первый аргумент командной строки (имя образа программы),
- `task` – номер задачи (1–10),
- `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 вещественных чисел и целое число n , и возвращающую ненулевое значение, если эта матрица симметрична (т.е. $a_{ij} = a_{ji}$), 0 в противном случае.
2. Написать функцию, получающую в качестве аргументов $n \times n$ матрицу A вещественных чисел и целое число n , и возвращающую след матрицы A .
3. Написать подпрограмму, получающую в качестве аргументов $n \times n$ матрицу A вещественных чисел и целое число n , и заменяющую матрицу A на ее транспонированную.
4. Написать подпрограмму, получающую в качестве аргументов $n \times n$ матрицу A вещественных чисел и целое число n , и заменяющую матрицу A на матрицу $(A + A^t)/2$, где A^t – транспонированная матрица A .
5. Написать подпрограмму, получающую в качестве аргументов $n \times n$ матрицу A вещественных чисел и целое число n , и заменяющую матрицу A на матрицу $(A - A^t)/2$, где A^t – транспонированная матрица A .
6. Написать подпрограмму, получающую в качестве аргументов $n \times m$ матрицу A вещественных чисел и целые числа n , m , i и j , и переставляющую в матрице A строки i и j местами. Номер строки и столбца начинается с 1.
7. Написать подпрограмму, получающую в качестве аргументов $n \times m$ матрицу A вещественных чисел и целые числа n , m , i и j , и переставляющую в матрице A столбцы i и j местами. Номер строки и столбца начинается с 1.
8. Написать подпрограмму, получающую в качестве аргументов $n \times m$ матрицу A вещественных чисел, целые числа n , m , i , j и вещественное число γ , и прибавляющую к j -й строке матрицы A i -ю строку, умноженную на число γ . Номер строки и столбца начинается с 1.
9. Написать подпрограмму, получающую в качестве аргументов $n \times m$ матрицу A вещественных чисел, вектор b длины m , вектор c длины n , целые числа n , m , и заменяющую вектор c на вектор, равный произведению матрицы A на вектор b ($c = Ab$).
10. Написать подпрограмму, получающую в качестве аргументов $n \times m$ матрицу A вещественных чисел, $m \times k$ матрицу B вещественных чисел, $n \times k$ матрицу C вещественных чисел, и целые числа n , m , k , и заменяющую матрицу C на матрицу, равную произведению матрицы A на матрицу B ($C = AB$).