

1. Дана матрица размера  $M \times N$ . Для каждого столбца матрицы с четным номером найти сумму его элементов. Условный оператор не использовать.
2. Дана целочисленная матрица размера  $M \times N$ . Найти номер первого из её столбцов, содержащих максимальное количество одинаковых элементов.
3. Дана матрица размера  $M \times N$ . Зеркально отразить её элементы относительно вертикальной оси симметрии матрицы (при этом поменяются местами столбцы с номерами 1 и  $N$ , 2 и  $N - 1$  и т. д.).
4. Дана квадратная матрица  $A$  порядка  $M$ . Найти среднее арифметическое элементов ее *побочной диагонали*, т.е. диагонали, содержащей элементы:  $A_{1M}$ ,  $A_{2M-1}$ ,  $A_{3M-2}$ , ...,  $A_{M1}$ .
5. Дана строка символов. Распечатать строку, удалив из неё все символы, не являющиеся буквами или цифрами и заменив каждую цифру двумя символами '\*.'
6. Дана строка, состоящая из русских слов, набранных заглавными буквами и разделённых пробелами (одним или несколькими). Вывести строку, содержащую эти же слова, разделённые одним пробелом и расположенные в алфавитном порядке.
7. Написать функцию `duration (T1, T2)`, возвращающую длительность промежутка времени  $dT$ , начиная от момента времени  $T1$  до  $T2$ . Для описания  $dT$ ,  $T1$ ,  $T2$  использовать структуру {часы, минуты, секунды}. Если  $T1 > T2$ , то считать, что момент времени  $T2$  относится к следующим суткам.
8. Массив записей именем PRICE содержит сведения о товарах {название товара; название магазина; цена товара}. Написать программу, обеспечивающую ввод с клавиатуры данных в массив PRICE и вывод на экран информации о товаре, название которого введено с клавиатуры. Если таких товаров нет, выдать на дисплей соответствующее сообщение.
9. Дан символьный файл F. Переписать в файл G сначала все нечётные его символы, а затем все чётные в обратном порядке.
10. Для хранения данных о планшетных сканерах описать структуру вида:

```

struct scan_info{
    char model[25]; // наименование модели
    int price;      // цена
    double x_size;  // горизонтальный размер области сканирования
    double y_size;  // вертикальный размер области сканирования
    int opttr;      // оптическое разрешение
    int grey;       // число градаций серого
};

```

Написать функцию, которая запрашивает количество сканеров, информация о которых будет вводиться, динамически выделяет память под массив структур соответствующего размера и заполняет его данными в режиме диалога (с клавиатуры). При этом имя сканера может содержать пробелы.

Написать функцию, которая записывает данный массив в создаваемый бинарный файл. Если цена сканера меньше 200, то данные об этом сканере в файл не записываются. Информация об остальных сканерах помещается в бинарный файл, причем сначала пишутся данные обо всех сканерах, имя которых начинается со строчной буквы, а затем — с прописной.

Структура файла: в первых четырех байтах размещается значение типа **long**, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах.

Привести пример программы, создающей файл с данными о сканерах и осуществляющей вывод на дисплей данных о требуемой записи (либо всех, либо по номеру).

Все необходимые данные для функций должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.