

Задание IV

Вариант 1

Текстовый файл `atom.dat` содержит базу данных по стабильным изотопам химических элементов от водорода до урана. Каждому изотопу соответствует одна запись (строка) в файле, которая содержит:

- Z* — атомный номер (заряд ядра, т.е. число протонов),
- A* — массовое число (число протонов + число нейтронов),
- El* — символ химического элемента,
- Mass* — атомная масса изотопа (в углеродных единицах, а.е.м.),
- Abund* — процентное содержание (abundance) в природной смеси изотопов данного элемента.

Напишите простейшую программу поиска информации в базе данных, которая по запросу пользователя выдает характеристики одного или нескольких изотопов.

Возможные формы запросов:

- `Ca` — выдать сведения обо всех стабильных изотопах кальция (в порядке увеличения их атомной массы),
- `Ca[40]` — выдать сведения об изотопе ^{40}Ca ,
- `Ca[]` — выдать сведения о наиболее распространенном изотопе кальция,
- `*[40]` — выдать сведения о любых изотопах с массовым числом 40 (т.е. ^{40}Ar , ^{40}K и ^{40}Ca) в порядке увеличения их атомного номера.

Форма вывода информации: общий заголовок, а далее одна или несколько строк, каждая из которых содержит данные об одном изотопе в следующем виде:

? Ca				
Изотоп	Z	N	Ат.масса	Содержание (%)
Ca[40]	20	20	39.962591	96.95
Ca[42]	20	22	41.958622	0.65
Ca[43]	20	23	42.958770	0.14

Ca[44]	20	24	43.955485	20.86
Ca[46]	20	26	45.953689	4.0e-003
Ca[48]	20	28	47.952532	0.19

? *[40]

Изотоп	Z	N	Ат.масса	Содержание (%)
Ar[40]	18	22	39.962383	99.60
K [40]	19	21	39.963999	1.2e-002
Ca[40]	20	20	39.962591	96.95

Здесь Z — число протонов, N — число нейтронов; процентное содержание выводится с двумя цифрами после точки, а если его величина меньше 0.1 — то с указанием порядка и двумя значащими цифрами (как показано выше).

Указания по программированию. Для хранения базы данных используйте массив структур вида:

```
struct isotope_data {
    int    z, a;
    char   el[3];
    double mass, abund;
}
```

Напишите отдельные функции для загрузки базы данных из файла, для грамматического разбора запроса пользователя, для поиска нужных записей по разным признакам в зависимости от типа запроса, для вывода данных об изотопе.

Вариант 2

Текстовый файл IVT.DAT содержит фрагмент базы данных ИВТАН-ТЕРМО¹ (термодинамические свойства индивидуальных веществ). Во вспомогательном файле ELEM.DAT находится особым образом упорядоченный список химических элементов, используемый при поиске информации. Структура файлов IVT.DAT и ELEM.DAT описана ниже.

¹Создана Институтом высоких температур Академии наук СССР (ИВТАН); отсюда название базы данных.

Напишите простейшую программу для получения из базы данных ИВТАНТЕРМО основных термодинамических характеристик веществ (C_p , Φ_T , S_T , $H_T - H_0$) при заданной температуре T . Предусмотрите следующие виды запросов пользователя:

- H 350.0 — выдать свойства всех веществ, в состав которых входит водород, для $T = 350$ К (8 веществ: H, H₂, H₂O, H₂O₂, HCl, H₂S, NH₃, N₂H₄).
- H+O 500.0 — выдать свойства веществ, в состав которых одновременно входят водород и кислород, для $T = 500$ К (2 вещества: H₂O, H₂O₂).
- =H2O 2400 — выдать свойства вещества с указанной формулой (H₂O) для $T = 2400$ К.

Пример вывода информации по второму запросу показан ниже.

? H+O 500

H2O Вода

T	Cp(T)	Φ(T)	S(T)	H(T)-H(0)
500.0	35.1321	172.7681	206.4069	16.8194

H2O2 Перекись водорода

T	Cp(T)	Φ(T)	S(T)	H(T)-H(0)
500.0	50.2300	217.1718	258.2382	20.5332

Главной термодинамической функцией, из которой получают все прочие свойства, служит приведенная энергия Гиббса $\Phi_T = -(G_T - H_0)/T$, где G_T — энергия Гиббса при температуре T , H_0 — энтальпия при абсолютном нуле. В базе данных ИВТАНТЕРМО приведенную энергию Гиббса Φ_T (в Дж/моль·К) приближенно представляют в виде

$$\Phi_T = c_1 + c_2 \ln x + \frac{c_3}{x^2} + \frac{c_4}{x} + c_5 x + c_6 x^2 + c_7 x^3, \quad \text{где } x = \frac{T}{10000}, \quad (1)$$

и хранят значения коэффициентов $c_1 - c_7$. Если выражение (1) не позволяет описать Φ_T с требуемой точностью во всем диапазоне температур (обычно от 300 до 20000 К), то этот диапазон разбивают на 2 или 3 части и для каждой из них задают свои коэффициенты.

Существуют уравнения, связывающие энтропию S_T , изменение энтальпии $H_T - H_0$ и теплоемкость при постоянном давлении C_p с приведенной энергией Гиббса Φ_T :

$$S_T = \Phi_T + T \frac{\partial \Phi}{\partial T}, \quad H_T - H_0 = T^2 \frac{\partial \Phi}{\partial T}, \quad C_p = \frac{\partial (H_T - H_0)}{\partial T}.$$

Если функция Φ_T имеет вид (1), то для величин S_T , $H_T - H_0$ и C_p получаются следующие выражения:

$$S_T = c_1 + c_2 (1 + \ln x) - \frac{c_3}{x^2} + 2c_5x + 3c_6x^2 + 4c_7x^3 \text{ Дж/}(\text{моль} \cdot \text{К}),$$

$$H_T - H_0 = 10 \cdot \left(c_2x - \frac{2c_3}{x} - c_4 + c_5x^2 + 2c_6x^3 + 3c_7x^4 \right) \text{ кДж/моль},$$

$$C_p = c_2 + \frac{2c_3}{x^2} + 2c_5x + 6c_6x^2 + 12c_7x^3 \text{ Дж/}(\text{моль} \cdot \text{К}).$$

Формат файла IVT.DAT

Файл IVT.DAT представляет собой фрагмент реальной базы данных и дан в «натуральном» виде. Далеко не вся содержащаяся в нем информация действительно нужна для этой задачи; те элементы данных, которые вам понадобятся, выделены в левой колонке жирным шрифтом.

Первая строка файла IVT.DAT содержит целое число — количество веществ в базе данных (16 в сокращенном варианте для задачи, более 2000 в реальной базе). Далее идут записи для разных веществ, отделенные друг от друга пустыми строками. Для каждого вещества данные хранятся в следующей форме:

Содержимое файла	Описание данных
20 Вода 1-В 17.09.80 H2O 3 2	Номер таблицы; название вещества Класс точности; дата составления таблицы Химическая формула вещества Номера химических элементов, из которых состоит вещество (по списку ELEM.DAT, 5 позиций на каждый номер, не более 7 номеров в списке)
1.8015200000E+01	Молекулярная масса (в атомных единицах)
-2.3891300000E+02	Теплота образования при 0 К, кДж/моль
-2.4181400000E+02	Теплота образования при 298.15 К, кДж/моль
1.1707000000E+01	Ядерная составляющая энтропии, Дж/(моль·К)
2.9815000000E+02	Стандартная температура T_0 (= 298.15 К)
3.3609000000E+01	Теплоемкость C_p при температуре T_0 , Дж/(моль·К)

(продолжение)	
1.5549200000E+02	Приведенная энергия Гиббса Φ при температуре T_0 , Дж/(моль·К)
1.8872400000E+02	Энтропия S при температуре T_0 , Дж/(моль·К)
9.9080000000E+00	Изменение энтальпии $H_{T_0} - H_0$, кДж/моль
-1.5187820000E+02	Десятичный логарифм константы равновесия $\lg K_p$ реакции образования из простых веществ при температуре T_0
3	Количество интервалов, на которых заданы коэффициенты уравнения (1). Ниже идут группы данных для этих интервалов, причем каждая группа занимает 9 строк:
298.15	Начальная температура интервала № 1
1500.00	Конечная температура интервала № 1
2.5398156738E+02	Коэффициент c_1 ур-я (1) на интервале № 1
2.7180023193E+01	Коэффициент c_2
9.8505709320E-04	Коэффициент c_3
-1.8467572331E-01	Коэффициент c_4
6.9049468994E+01	Коэффициент c_5
2.8828838348E+01	Коэффициент c_6
-1.1553672790E+02	Коэффициент c_7
1500.00	Начальная температура интервала № 2
6000.00	Конечная температура интервала № 2
2.6391625977E+02	Коэффициент c_1 ур-я (1) на интервале № 2
3.5357559204E+01	Коэффициент c_2
.	и т. д.
-3.9762330055E+00	Коэффициент c_7 ур-я (1) на интервале № 3 (конец группы данных для соединения H ₂ O).

Формат файла *ELEM.DAT*

Файл ELEM.DAT содержит символы и массовые числа химических элементов, входящих в состав соединений, включенных в базу данных. Элементы упорядочены особым образом: на первом месте электрон, затем кислород, далее изотопы водорода, а остальные элементы расположены по подгруппам периодической системы. Несколько первых строк этого

файла показаны ниже:

е	0
О	16
Н	1
Д	2
Т	3
Ф	19

Нумерация записей в файле ELEM.DAT используется при поиске нужных соединений в IVT.DAT: в 4-й строке группы данных о каждом соединении находится список номеров элементов. Так, в приведенных выше данных о воде H₂O 4-я строка содержит числа 3 и 2 — это порядковые номера записей об элементах Н и О в файле ELEM.DAT.

Обработывая запросы первых двух типов (с поиском соединений по одному или нескольким элементам) программа прежде всего должна найти указанные химические символы в ELEM.DAT, а уже затем, узнав их порядковые номера, искать информацию о соединениях в файле IVT.DAT. Запросы третьего типа (поиск по химической формуле соединения) не требуют обращения к файлу ELEM.DAT.

Вариант 3

Напишите программу, которая читает текст (на русском языке, без переносов) из указанного файла и форматирует его следующим образом:

- выравнивает правую границу текста, т.е. добивается, чтобы все строки за исключением последней строки абзаца имели одинаковую ширину (длинные строки переносятся по границе слов, а короткие расширяются путем вставки дополнительных пробелов между словами по изложенным ниже правилам);
- отделяет абзацы текста друг от друга пустой строкой;
- делает отступ (несколько пробелов) в начале первой строки каждого абзаца («красная строка»).

В исходном тексте абзацы могут быть разделены пустыми строками либо идти подряд без промежутков; в последнем случае признаком

нового абзаца служит красная строка (2 и более пробелов в начале строки). Несколько идущих подряд пустых строк необходимо сохранять (бывает нужен увеличенный промежуток между разделами текста). Слова исходного текста разделены одним или несколькими пробелами и/или знаками табуляции. В пустой строке могут находиться символы пустых промежутков (пробелы и знаки табуляции), но ничего кроме них.

Правила выравнивания строк:

- в первую очередь дополнительные пробелы вставляются в конце предложений, т. е. после знаков препинания «.», «!», «?»;
- во вторую очередь пробелы добавляются после знаков препинания в середине предложений, т. е. после «,», «;», «:» и вокруг тире «—»;
- в последнюю очередь добавляются пробелы между словами, не разделенными какими-либо знаками препинания.

Названия входного и выходного файлов, а также параметры форматирования (ширина строк, величина отступа в красной строке) должны задаваться в командной строке при запуске программы, например:

```
formtext -w:72 -i:5 file1.txt file2.txt
```

Здесь `formtext` — название программы, `-w:72` задает ширину текста (width), `-i:5` задает величину отступа в красной строке (indent), исходный текст следует читать из файла `file1.txt`, результат записать в файл `file2.txt`. Опции `-w` и `-i` могут отсутствовать; в этом случае должны быть приняты значения по умолчанию: ширина строки 65 позиций, величина отступа 5 позиций. В случае `-i:0` красной строки в начале абзаца нет.

Образцы текстов для форматирования даны в файлах `text43_1.txt` и `text43_2.txt`.

Вариант 4

Напишите программу, которая читает текст (на русском языке) из указанного файла и обрабатывает его следующим образом:

- убирает переносы, если они есть, «склеивая» части слова в одно целое;

- форматирует строки так, чтобы они не выходили за пределы заданной ширины текста (при этом выравнивать правую границу не нужно). Промежутки между словами должны быть минимальными (ровно один пробел), а слова разрешено переносить со строки на строку только целиком;
- отделяет абзацы текста друг от друга пустой строкой;
- делает отступ (несколько пробелов) в начале первой строки каждого абзаца («красная строка»).

В исходном тексте абзацы могут быть разделены пустыми строками либо идти подряд без промежутков; в последнем случае признаком нового абзаца служит красная строка (2 и более пробелов либо знак табуляции в начале строки). Несколько идущих подряд пустых строк необходимо сохранять (их используют, когда бывает нужен увеличенный промежуток между разделами текста). Слова исходного текста разделены одним или несколькими пробелами и/или знаками табуляции. В пустой строке могут находиться символы пустых промежутков (пробелы и знаки табуляции), но ничего кроме них.

Названия входного и выходного файлов, а также параметры форматирования (ширина текста, величина отступа в красной строке) должны задаваться в командной строке при запуске программы, например:

```
formtext -w:72 -i:5 file1.txt file2.txt
```

Здесь `formtext` — название программы, `-w:72` задает ширину текста (width), `-i:5` задает величину отступа в красной строке (indent), исходный текст следует читать из файла `file1.txt`, результат записать в файл `file2.txt`. Опции `-w` и `-i` могут отсутствовать; в этом случае должны быть приняты значения по умолчанию: ширина текста 65 позиций, величина отступа 5 позиций. В случае `-i:0` красной строки в начале абзаца нет.

Образцы текстов для форматирования даны в файлах `text44_1.txt` и `text44_2.txt`.

Вариант 5

Некоторое время назад на компьютерах применяли не 8-, а 7-битную кодировку символов (старший, 8-й бит байта использовали для контро-

ля ошибок путем дополнения кода до нечетного числа единиц). Поэтому можно было изобразить не 256, а всего лишь 128 разных символов. Этого недостаточно для совместного представления латинского и русского алфавитов, поскольку кроме букв нужны еще цифры, знаки препинания и прочие специальные символы. По этой причине в России была распространена кодировка, в которой отсутствовали строчные буквы (текст записывали только заглавными буквами), и при этом не различали русские и латинские буквы, совпадающие или похожие по начертанию (такие, как А, В, С, Е, Н, К, М, О, Р, Т, Х, У). Текстовые файлы, созданные в те времена и сохранившие «укороченный» русский алфавит, можно встретить до сих пор. Например, в таком виде распространялся первый любительский перевод² книги Б. Кернигана и Д. Ричи «Язык программирования С».

Напишите программу, которая читает текст на русском языке в кодировке `sr866` (DOS) из одного файла и копирует его в другой файл, выполняя преобразование к русскому алфавиту. При этом структуру текста (т. е. расположение слов, разбиение на строки) нужно сохранить. Заглавные буквы (за исключением первой буквы каждого предложения) необходимо превратить в строчные. Начало предложения определяется по знакам препинания «.», «!», «?» в конце предыдущего слова (включая комбинации «...», «!...», «?!» и т. д.), причем после этих знаков может стоять тире (возможно, отделенное пробелом), что часто встречается при передаче прямой речи.

Примечание. В научно-технических текстах встречаются слова, записанные латинскими буквами, и в них преобразование латинских букв в русские неуместно. Поэтому не следует преобразовывать те слова, где обнаружены латинские буквы, не имеющие русского аналога (например, F, G, R, S, W и т. д.). Однако полностью исключить нежелательные преобразования при автоматической обработке текста невозможно.

Названия входного и выходного файлов, а также параметры обработки должны задаваться в командной строке при запуске программы, например:

```
recode -u -y file1.txt file2.txt
```

Здесь `recode` — название программы, `-u` запрещает перевод заглавных букв (uppercase letters) в строчные, `-y` говорит, что латинская буква Y

²Перевод В. В. Чуклова, появившийся в 1982 году, спустя 4 года после выхода первого английского издания.

не совпадает с русской У. Образец текста для обработки дан в файле text45.txt.

Указания по программированию. Замену кодировки или систематическую замену одних символов другими обычно выполняют следующим образом. Создают таблицу перекодировки, т.е. массив символов, заполненный так, что код исходного символа служит индексом элемента таблицы, а в самом элементе хранится замена, т.е. код, который нужно подставить вместо исходного. Кодировку cp866 (DOS) можно найти в файле codetable.dos.

Напишите следующие функции:

`int latrus (int c)` — для заглавной латинской буквы `c`, совпадающей по начертанию с одной из русских букв, возвращает соответствующую русскую букву, а все прочие символы возвращает не измененными.

`int rustolower (int c)` — для заглавной русской буквы `c` возвращает соответствующую строчную букву, а все прочие символы возвращает не измененными. (В языке C существует стандартная функция `tolower (c)`, но она не умеет работать с русскими буквами.)

Подумайте, какие еще функции удобно ввести, чтобы сделать программу более простой и понятной (например, чтение очередного слова/строки/абзаца, выделение одного слова и анализ символов, из которых оно состоит, обработка параметров командной строки и т.д.)