**2.1 Памятка по работе со строками**

**Памятка по кодированию слов различной длины**

**Кодирование слова длиной 8 бит:**

*Обычное описание:*  
При использовании 8-битного кода для представления одного символа (или байта), мы можем закодировать до 28 = 256 различных символов. Это позволяет использовать такие наборы символов, как ASCII, где каждый символ представлен одним байтом. Эмодзи-символы, такие как 😊, не могут быть закодированы в рамках этого формата, так как они требуют больше места.

*Пример расчета:*  
Допустим, нам нужно закодировать символ "A". В таблице ASCII этот символ имеет значение 65. В двоичной системе счисления это число представляется как:

6510 = 010000012

Таким образом, символ "A" занимает ровно 8 бит (один байт).

**Кодирование слова длиной 16 бит**

*Обычное описание:*  
16-битный код (два байта) используется для представления расширенных наборов символов, таких как Unicode (UTF-16). В этом случае возможно представить до 216 = 65 536 различных символов, что достаточно для большинства современных языков мира, а также некоторых базовых эмодзи, таких как 😊.

*Пример расчета:*  
Рассмотрим символ "😊" (улыбающаяся рожица). В кодировке UTF-16 этот символ имеет значение 263A в шестнадцатеричной системе. Переведем это значение в двоичную систему:

263A16​= 00100110001110102​

Как видно, этот символ занимает 16 бит (два байта).

**Кодирование слова длиной 32 бита**

*Обычное описание:*   
32-битная кодировка (четыре байта) может использоваться для поддержки всех символов Unicode, включая редко используемые символы и сложные эмодзи, такие как 🎉 (конфетти-баллон) или 🏳️‍🌈 (радужный флаг).

*Пример расчета:*   
Возьмем символ "🏳️‍🌈" (радужный флаг). В кодировке UTF-32 этот символ имеет значение 1F3F3 FE0F 200D 1F308 в шестнадцатеричной системе. Переведем каждое значение в двоичную систему:

1F3F316 = 0001 1111 0011 11112

FE0F16 = 1111 1110 0000 11112

200D16 = 0010 0000 0000 11012

1F30816 = 0001 1111 0011 10002

Эти четыре значения вместе занимают 128 бит (4 байта), что соответствует полной длине символа "🏳️‍🌈".

Таким образом, выбор длины слова зависит от того, какие символы необходимо поддерживать, включая эмодзи.

**2.2 Кодирование слова длиной 8 бит**

**Задание 1 (ручной способ):**

Определить вычеркнутое слово в следующей последовательности:

«Обь, Лена, Волга, Москва, Макензи, Амазонка  - реки».

Известно, что каждый символ кодируется **8 битами**. При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **8 байтов** меньше, чем размер исходного предложения.

**Решение:**

Для решения задачи используем следующую логику:

1. *Узнаем, сколько байтов занимает каждый символ.*  
   Так как 1 символ кодируется 8 битами, это эквивалентно 1 байту.
2. *После удаления слова из последовательности удаляются также пробел и запятая.*  
   Это значит, что размер удаляемого блока увеличивается на 2 символа.
3. *Новый размер последовательности оказался на 8 байтов меньше исходного,* значит длина удаленного слова, включая пробел и запятую, составляет 8 байтов.

**1. Вычислим длину удаленного слова:**

* Длина удаленного блока = длина слова + 2 (за счет пробела и запятой).
* **Длина слова = длина удаленного блока - 2 = 8 - 2 = 6 байтов.**

**2. Вычисление длины слов в предложении:**

* «Обь» — 3 символа (3 байта).
* «Лена» — 4 символа (4 байта).
* «Волга» — 5 символов (5 байтов).
* **«Москва» — 6 символов (6 байтов).**
* «Макензи» — 7 символов (7 байтов).
* «Амазонка» — 8 символов (8 байтов).

Единственное слово, длина которого равна 6 байтам, — это «Москва».

**Ответ:***Москва.*

**2.3 Кодирование слова длиной 16 бит**

**Шаг 1**

**Задание 1 (ручной способ):**

Определить размер предложения в кодировке Unicode:

«Я к вам пишу - чего же боле? Что я могу еще сказать?».

Известно, что каждый символ в кодировке Unicode кодируется **16 битами**. Размер предложения необходимо вычислить в **байтах**.

**Решение:**

Для решения задачи определим количество символов в предложении и используем информацию о том, что каждый символ в Unicode кодируется 16 битами.

#### ****1. Переводим биты в байты:****

16 бит = 2 байта на символ

#### ****2. Подсчитаем количество символов в предложении:****

Предложение:

«Я к вам пишу — чего же боле? Что я могу еще сказать?»

Посчитаем все символы, включая пробелы и знаки пунктуации:

52 символа

#### ****3. Вычислим общий размер:****

Общий размер предложения в байтах по следующей формуле:

Размер = Количество символов × Размер одного символа (в байтах)

52 × 2 = 104 байта

**Ответ:**104 байта.

**Шаг 2**

**Задание 1.**

Написать программу на **C++**, которая определяет размер предложения в кодировке *Unicode*:

«Я к вам пишу - чего же боле? Что я могу еще сказать?».

Известно, что каждый символ в кодировке *Unicode* кодируется **16 битами**. Размер предложения необходимо вычислить в **байтах**.

**Решение:**

Чтобы решить задачу, создадим программу, которая будет вычислять вес в байтах исходного предложения при условии, что каждый символ кодируется 16 битами.

**Листинг:**

#include <iostream>

#include <string>

using namespace std;

int main() {

    // Исходная строка

    string s = u8"Я к вам пишу — чего же боле? Что я могу еще сказать?"; // UTF-8 строка

    // Количество бит на символ

    const unsigned short bns = 16;          // 16 бит на символ

    const float bans = static\_cast<float>(bns) / 8; // переводим биты в байты

    // Вычисляем размер строки в байтах

    size\_t bytes = [s.length()](s.length()) \* bans;

    cout << "Размер предложения в байтах: " << static\_cast<int>(bytes) << endl;

    return 0;

}

**Пошаговое объяснение кода**

**1. Исходная строка:**

string s = u8"Я к вам пишу — чего же боле? Что я могу еще сказать?"; // UTF-8 строка

   Мы используем `u8` префикс перед строкой, чтобы указать компилятору, что строка представлена в формате UTF-8. Таким образом, мы можем корректно считать длину строки в символах.

**2. Количество бит на символ:**

const unsigned short bns = 16;           // 16 бит на символ

   const float bans = static\_cast<float>(bns) / 8; // переводим биты в байты

   Здесь мы задаём количество бит на символ (`bns`) и рассчитываем количество байтов на символ (`bans`), используя деление.

**3. Вычисление размера строки в байтах:**

size\_t bytes = [s.length()](s.length()) \* bans;

   Функция `length()` возвращает количество символов в строке, а затем мы умножаем эту величину на количество байт на символ.

**4. Вывод результата:**

cout << "Размер предложения в байтах: " << static\_cast<int>(bytes) << endl;

   Выводим результат, предварительно приведя его к целочисленному типу.

**Ответ:***104 байта.*

**2.4 Кодирование слова длиной 32 бита**

**Шаг 1**

**Задание 1 (ручной способ):**

Определить вычеркнутое слово в следующей последовательности:

«Врач, актер, акушер, генетик, издатель, кардиолог - профессии»

Известно, что каждый записан в кодировке UTF-32 и кодируется **32 битами**. При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **44 байта** меньше, чем размер исходного предложения.

**Pешение:**

Для решения задачи используем следующие шаги:

**1. Основные данные:**

* **UTF-32**: каждый символ кодируется 32 битами = 4 байтами.
* После удаления слова также удаляются **пробел** и **запятая**, то есть дополнительно 2 символа (по 4 байта каждый).
* Уменьшение размера последовательности составляет 44 байта.

Это означает, что общая длина удаленного блока составляет 44 байта\text{44 байта}.

**2. Определение длины удаленного слова:**

* Общая *длина удаленного блока = длина слова (в байтах) + длина пробела и запятой (в байтах).*
* *Длина слова (в байтах) = общая длина удаленного блока - длина пробела и запятой*.

Длина пробела и запятой: 2 × 4 = 8 байт.  
Длина слова: 44 − 8 = 36 байт.

Поскольку в UTF-32 каждый символ занимает 4 байта, длина слова в символах:

36 : 4 = 9 символов

**3. Поиск слова длиной 9 символов:**

Из списка слов:

* «Врач» — 4 символа.
* «актер» — 5 символов.
* «акушер» — 7 символов.
* «генетик» — 7 символов.
* «издатель» — 8 символов.
* **«кардиолог» — 9 символов.**

**Ответ:***кардиолог.*

**Шаг 2**

**Задание 2 (на языке С++):**

Написать программу на **С++**, которая определяет вычеркнутое слово в следующей последовательности:

*«Врач, актер, акушер, генетик, издатель, кардиолог - профессии».*

Известно, что каждый записан в кодировке UTF-32 и кодируется **32 битами**. При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **44 байта** меньше, чем размер исходного предложения.

**Решение:**

Чтобы решить задачу, создадим программу, которая будет вычислять удаленное слово после его вычеркивания из исходной последовательности при условии, что каждый символ кодируется **32 битами**, а вес сроки после удаления слова уменьшился на **44 байта**.

#include <iostream>

#include <string>

#include <vector>

int main() {

    // Исходная строка

    std::string str = "Врач, актер, акушер, генетик, издатель, кардиолог";

    // Параметры

    const unsigned short vs = 32;            // Количество бит на символ

    const float vsb = static\_cast<float>(vs) / 8; // Вес одного символа в байтах

    const float menshe = 44;                 // Уменьшение на 44 байта

    const float ksymbols = menshe / vsb - 2; // Количество символов, которые были удалены

    // Разделяем строку на слова

    std::istringstream iss(str);

    std::vector<std::string> words((std::istream\_iterator<std::string>(iss)), std::istream\_iterator<std::string>());

    // Поиск удалённого слова

    for (const auto& word : words) {

        if (word.size() == ksymbols) {

            std::cout << "Удалённое слово: " << word << std::endl;

            break;

        }

    }

    return 0;

}

**Пошаговое объяснение кода**

**1. Исходная строка:**

std::string str = "Врач, актер, акушер, генетик, издатель, кардиолог";

   Эта строка содержит список профессий, разделённых запятыми.

**2. Параметры:**

const unsigned short vs = 32;            // Количество бит на символ

   const float vsb = static\_cast<float>(vs) / 8; // Вес одного символа в байтах

   const float menshe = 44;                 // Уменьшение на 44 байта

   const float ksymbols = menshe / vsb - 2; // Количество символов, которые были удалены

   Эти параметры определяют условия задачи: количество бит на символ, вес одного символа в байтах, уменьшение в байтах и расчёт количества символов, которые были удалены.

3.Разделение строки на слова:

std::istringstream iss(str);

   std::vector<std::string> words((std::istream\_iterator<std::string>(iss)), std::istream\_iterator<std::string>());

   Используется потоковая обработка через `std::istringstream` для разделения строки на отдельные слова. Результат сохраняется в вектор `words`.

**4. Поиск удалённого слова:**

for (const auto& word : words) {

       if (word.size() == ksymbols) {

           std::cout << "Удалённое слово: " << word << std::endl;

           break;

       }

   }

   Цикл проходит по каждому слову в векторе `words`, проверяет соответствие длины слова количеству символов, которые были удалены. Если совпадение найдено, слово выводится на экран, и цикл останавливается.

**Ответ:***кардиолог.*

**2.5 Разные задачи**

**Шаг 1**

**Задание 1 (ручной способ):**

Необходимо вычислить информационный вес рассказа, если он содержит**8 страниц**, на каждой странице **40 строк**, в каждой строке **48 символов**. Вес вычислить КБайтах, при это известно, что информационный вес одного символа составляет **8 бит**.

**Решение:**

Для вычисления информационного веса рассказа выполним следующие шаги:

#### ****1. Вычислим общее количество символов в рассказе:****

Общее количество символов = Количество страниц × Количество строк на странице × Количество символов в строке

Подставляем значения:

8 × 40 × 48 = 15 360 символов

#### ****2. Вычислим общий объем информации в битах:****

Объем в битах = Общее количество символов × Информационный вес одного символа (в битах)

Подставляем:

15 360 × 8=122 880 бит

#### ****3. Преобразуем объем информации в байты (1 байт = 8 бит):****

Объем в байтах = Объем в битах : 8

Подставляем:

122 880 : 8 = 15 360 байт

#### ****4. Преобразуем объем информации в килобайты (1 КБ = 1024 байта):****

Объем в КБ = Объем в байтах : 1024

Подставляем:

15 360 : 1024 = 15 Кб

**Ответ:** 15 Кб.

**Шаг 2**

**Задание 1 (на языке C++):**

Необходимо вычислить информационный вес рассказа с помощью языка программирования **C++**, если он содержит**8 страниц**, на каждой странице **40 строк**, в каждой строке **48 символов**. Вес вычислить КБайтах, при это известно, что информационный вес одного символа составляет **8 бит**.

**Решение:**

Чтобы решить задачу, создадим программу, которая будет вычислять информационный вес рассказа в КБайтах, при условии вес одного символа составляет **8 бит.**

**Листинг:**

#include <iostream>

using namespace std;

int main() {

int stranic = 8; // Количество страниц в документе

int strok = 40; // Количество строк на странице

int stroka = 48; // Количество символов в строке

int vs = 8; // Один символ кодируется 8 битами

double vsb = vs / [8.0](8.0); // Вес одного символа в байтах

double V = stranic \* strok \* stroka \* vsb; // Информационный вес документа в байтах

V /= (1 << 10); // Переводим байты в килобайты (2^10)

cout << static\_cast<int>(V) << endl; // Выводим результат на экран

return 0;

}

**Пошаговое объяснение кода**

**1. Подключение библиотеки ‘iostream’**

#include <iostream>

Эта строка подключает стандартную библиотеку ввода-вывода (‘iostream’), чтобы мы могли использовать функции вывода на экран, такие как ‘cout’.

**2. Пространство имен ‘std’**

using namespace std;

Это позволяет нам использовать функции и классы из пространства имен ‘std’ без необходимости каждый раз указывать ‘std::’. Например, теперь мы можем писать ‘cout’, а не ‘std::cout’.

**3. Основная функция программы — ‘main()’**

int main() {

Функция ‘main()’ является точкой входа в программу. Она возвращает целое число (‘int’), которое обозначает статус завершения программы (обычно 0 означает успешное завершение).

**4. Объявление переменных**

int stranic = 8;           // Количество страниц в документе

    int strok = 40;             // Количество строк на странице

    int stroka = 48;            // Количество символов в строке

    int vs = 8;                 // Один символ кодируется 8 битами

Здесь объявляются четыре целочисленные переменные:  
- ‘stranic’: количество страниц в документе,  
- ‘strok’: количество строк на одной странице,  
- ‘stroka’: количество символов в одной строке,  
- ‘vs’: количество битов, необходимых для хранения одного символа (в данном случае 8 бит).

**5. Вычисляем вес одного символа в байтах**

double vsb = vs / [8.0](8.0);     // Вес одного символа в байтах

Переменная ‘vsb’ хранит вес одного символа в байтах. Поскольку один байт равен 8 битам, делим ‘vs’ на 8. Мы используем тип ‘double’, чтобы избежать потери точности при делении целых чисел.

**6. Вычисляем общий объем документа в байтах**

double V = stranic \* strok \* stroka \* vsb;   // Информационный вес документа в байтах

Мы вычисляем общий объем документа в байтах, умножая количество страниц на количество строк на каждой странице, на количество символов в строке и на вес одного символа в байтах.

**7. Преобразуем байты в килобайты**

V /= (1 << 10);            // Переводим байты в килобайты (2^10)

Для перевода байтов в килобайты делим общее количество байтов на 210210 (1024). Это делается с помощью сдвига влево на 10 позиций, что эквивалентно умножению на 1024.

**8. Вывод результата на экран**

cout << static\_cast<int>(V) << endl;  // Выводим результат на экран

Используя оператор ‘<<‘, выводим значение переменной ‘V’ на экран. Поскольку ‘V’ имеет тип ‘double’, мы приводим его к типу ‘int’ с помощью оператора приведения типа ‘static\_cast<int>‘, чтобы получить целую часть числа. Функция ‘endl’ добавляет перевод строки после вывода значения.

**9. Завершение программы**

return 0;

}

Возвращаем 0, что сигнализирует операционной системе о нормальном завершении программы.

**Ответ:** *15 Кб.*

**2.6 Примеры по кодированию слов длиной 8 бит**

**Шаг 1**

**Задание 1:**

Определить вычеркнутое слово в следующей последовательности:

«Солнце, Луна, Земля, Марс, Венера, Юпитер – планеты».

Известно, что каждый символ кодируется **8 битами.** При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **7 байтов** меньше, чем размер исходного предложения.

**Решение:**

Для решения задачи используем следующую логику:

1. Узнаем, сколько байтов занимает каждый символ. Так как 1 символ кодируется 8 битами, это эквивалентно 1 байту.
2. После удаления слова из последовательности удаляются также пробел и запятая. Это значит, что размер удаляемого блока увеличивается на 2 символа.
3. Новый размер последовательности оказался на 7 байтов меньше исходного, значит длина удаленного слова, включая пробел и запятую, составляет 7 байтов.

**1. Вычислим длину удаленного слова:**

* Длина удаленного блока = длина слова + 2 (за счет пробела и запятой).
* Длина слова = длина удаленного блока - 2 = 7 - 2 = 5 байтов.

**2. Вычисление длины слов в предложении:**

* «Солнце» – 6 символа (6 байта).
* «Луна» – 4 символа (4 байта).
* «Земля» – 5 символов (5 байтов).
* «Марс» – 4 символа (4 байта).
* «Венера» – 6 символов (6 байтов).
* «Юпитер» – 6 символов (6 байтов).

Единственное слово, длина которого равна 5 байтам, – это «Земля».

**Ответ:**Земля

**Шаг 2**

**Задание 2:**

Определить вычеркнутое слово в следующей последовательности:

«Аквамарин, Синий, Зеленый, Алый, Белый, Фуксия».

Известно, что каждый символ кодируется **8 битами.** При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **6 байтов** меньше, чем размер исходного предложения.

**Решение:**

Для решения задачи используем следующую логику:

1. Узнаем, сколько байтов занимает каждый символ. Так как 1 символ кодируется 8 битами, это эквивалентно 1 байту.
2. После удаления слова из последовательности удаляются также пробел и запятая. Это значит, что размер удаляемого блока увеличивается на 2 символа.
3. Новый размер списка уменьшился на 6 байт после удаления слова “Синий”, значит, его длина составляет 6 байт.

**1. Вычислим длину удаленного слова:**

* Длина удаленного блока = длина слова + 2 (за счет пробела и запятой).
* Длина слова = длина удаленного блока - 2 = 6 - 2 = 4 байтов.

**2. Вычисление длины слов в предложении:**

* “Аквамарин” - 8 байтов
* “Синий” - 5 байтов
* “Зеленый” - 7 байтов
* “Алый” - 4 байта
* “Белый” - 5 байта
* “Фуксия” - 6 байта

**Ответ:**Алый

**Шаг 3**

**Задание 3:**

Определить вычеркнутое слово в следующей последовательности:

«Москва, Сочи, Новгород, Владивосток – города России».

Известно, что каждый символ кодируется **8 битами.** При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **8 байтов** меньше, чем размер исходного предложения.

#### ****Решение:****

1. Удаленный блок (слово + пробел + запятая) = 8 байтов.
2. Длина слова = 8 - 2 = 6 символов.
3. Анализ слов:
   * «Москва» – 6 символов,
   * «Сочи» – 4 символа,
   * «Новгород» – 8 символов,
   * «Владивосток» – 11 символов.

**Ответ:**Москва

**Шаг 4**

**Задание 4:**

Определить вычеркнутое слово в следующей последовательности:

«Яндекс, Google, Samsung, Microsoft – IT-компании».

Известно, что каждый символ кодируется **8 битами.** При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **9 байтов** меньше, чем размер исходного предложения.

#### ****Решение:****

1. Удаленный блок = 9 байтов.
2. Длина слова = 9 - 2 = 7 символов.
3. Проверка слов:
   * «Яндекс» – 6 символов,
   * «Google» – 6 символов,
   * «Samsung» – 7 символов,
   * «Microsoft» – 9 символов.

**Ответ:**Samsung

**Шаг 5**

**Задание 5:**

Определить вычеркнутое слово в следующей последовательности:

«Ромашка, Тюльпан, Гвоздика, Пион – цветы».

Известно, что каждый символ кодируется **8 битами.** При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **9 байтов** меньше, чем размер исходного предложения.

#### ****Решение:****

1. Удаленный блок = 9 байтов.
2. Длина слова = 9 - 2 = 7 символов.
3. Анализ слов:
   * «Ромашка» – 7 символов,
   * «Тюльпан» – 7 символов,
   * «Гвоздика» – 8 символов,
   * «Пион» – 4 символа.  
     **Ответ:** Тюльпан (первое подходящее).

**Ответ:**Samsung

**Шаг 6**

**Задание 6:**

Определить вычеркнутое слово в следующей последовательности:

«Молоток, Отвертка, Плоскогубцы, Дрель, Пила, Гвоздь – инструменты».

Известно, что каждый символ кодируется **8 битами.** При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **10 байтов** меньше, чем размер исходного предложения.

#### ****Решение:****

1. Удаленный блок = 10 байтов.
2. Длина слова = 10 - 2 = 8 символов.
3. Анализ слов:
   * «Молоток» – 7 символов,
   * «Отвертка» – 8 символов,
   * «Плоскогубцы» – 11 символов,
   * «Дрель» – 5 символов,
   * «Пила» – 4 символа,
   * «Гвоздь» – 6 символов.

**Ответ:**Отвертка

**Шаг 7**

**Задание 7:**

Определить вычеркнутое слово в следующей последовательности:

«Альфа, Бета, Гамма, Дельта, Эпсилон, Омега – греческие буквы».

Известно, что каждый символ кодируется **8 битами.** При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **6 байтов** меньше, чем размер исходного предложения.

#### ****Решение:****

1. Удаленный блок = 6 байтов.
2. Длина слова = 6 - 2 = 4 символа.
3. Проверка слов:
   * «Альфа» – 5 символов,
   * «Бета» – 4 символа,
   * «Гамма» – 5 символов,
   * «Дельта» – 6 символов,
   * «Эпсилон» – 7 символов,
   * «Омега» – 5 символов.

**Ответ:** Бета

**Шаг 8**

**Задание 8:**

Определить вычеркнутое слово в следующей последовательности:

«Кофе, Чай, Сок, Молоко, Вода, Лимонад – напитки».

Известно, что каждый символ кодируется **8 битами.** При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **5 байтов** меньше, чем размер исходного предложения.

#### ****Решение:****

1. Удаленный блок = 5 байтов.
2. Длина слова = 5 - 2 = 3 символа.
3. Анализ слов:
   * «Кофе» – 4 символа,
   * «Чай» – 4 символа,
   * «Сок» – 3 символа,
   * «Молоко» – 6 символов,
   * «Вода» – 4 символа,
   * «Лимонад» – 7 символов.

**Ответ:** Сок

**Шаг 9**

**Задание 9:**

Определить вычеркнутое слово в следующей последовательности:

«Дуб, Береза, Сосна, Клён, Баобаб, Эвкалипт – деревья».

Известно, что каждый символ кодируется **8 битами.** При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **6 байтов** меньше, чем размер исходного предложения.

#### ****Решение:****

1. Длина слова = 6 - 2 = 4 символа
2. Анализ:
   * «Дуб» – 3
   * «Береза» – 6
   * «Сосна» – 5
   * «Клён» – 4
   * «Баобаб» – 6
   * «Эвкалипт» – 8

**Ответ:** Клён

**Шаг 10**

**Задание 10:**

Определить вычеркнутое слово в следующей последовательности:

«Волк, Лиса, Медведь, Барсук, Сурок, Носорог – животные».

Известно, что каждый символ кодируется **8 битами.** При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **8 байтов** меньше, чем размер исходного предложения.

#### ****Решение:****

1. Длина слова = 8 - 2 = 6 символов
2. Анализ:
   * «Волк» – 4
   * «Лиса» – 4
   * «Медведь» – 7
   * «Барсук» – 6
   * «Сурок» – 5
   * «Носорог» – 7

**Ответ:** Барсук

**Шаг 11**

**Задание 11:**

Определить вычеркнутое слово в следующей последовательности:

«Сапфир, Рубин, Топаз, Изумруд, Гранат, Янтарь – драгоценные камни».

Известно, что каждый символ кодируется **8 битами.** При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **9 байтов** меньше, чем размер исходного предложения.

#### ****Решение:****

1. Длина слова = 9 - 2 = 7 символов
2. Анализ:
   * «Сапфир» – 6
   * «Рубин» – 5
   * «Топаз» – 5
   * «Изумруд» – 7
   * «Гранат» – 6
   * «Янтарь» – 6

**Ответ:** Изумруд

**Шаг 12**

**Задание 12:**

Определить вычеркнутое слово в следующей последовательности:

«Скрипка, Арфа, Флейта, Виолончель, Гитара, Балалайка – музыкальные инструменты».

Известно, что каждый символ кодируется **8 битами.** При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **10 байтов** меньше, чем размер исходного предложения.

#### ****Решение:****

1. Длина слова = 10 - 2 = 8 символов
2. Анализ:
   * «Скрипка» – 7
   * «Арфа» – 4
   * «Флейта» – 6
   * «Виолончель» – 10
   * «Гитара» – 6
   * «Балалайка» – 9

**Ответ:** Виолончель

**Шаг 13**

**Задание 13:**

Определить вычеркнутое слово в следующей последовательности:

«Пёс, Кошка, Ёж, Черепаха, Попугай, Рыбки – питомцы»

Известно, что каждый символ кодируется **8 битами.** При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **5 байтов** меньше, чем размер исходного предложения.

#### ****Решение:****

1. Длина слова = 5 - 2 = 3 символа
2. Анализ:
   * «Пёс» – 3
   * «Кошка» – 5
   * «Ёж» – 2
   * «Черепаха» – 8
   * «Попугай» – 7
   * «Рыбки» – 5

**Ответ:** Пёс

**Шаг 14**

**Задание 14:**

Определить вычеркнутое слово в следующей последовательности:

«Январь, Февраль, Март, Апрель, Май, Июнь – месяцы»

Известно, что каждый символ кодируется **8 битами.** При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **5 байтов** меньше, чем размер исходного предложения.

#### ****Решение:****

1. Длина слова = 5 - 2 = 3 символа
2. Анализ:
   * «Январь» – 6
   * «Февраль» – 7
   * «Март» – 4
   * «Апрель» – 6
   * «Май» – 3
   * «Июнь» – 4

**Ответ:** Май

**Шаг 15**

**Задание 15:**

Определить вычеркнутое слово в следующей последовательности:

«Роза, Лилии, Пионы, Гвоздика, Нарцисс, Фиалка – цветы»

Известно, что каждый символ кодируется **8 битами.** При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **8 байтов** меньше, чем размер исходного предложения.

#### ****Решение:****

1. Длина слова = 8 - 2 = 6 символов
2. Анализ:
   * «Роза» – 4
   * «Лилии» – 5
   * «Пионы» – 5
   * «Гвоздика» – 8
   * «Нарцисс» – 7
   * «Фиалка» – 6

**Ответ:** Фиалка

**Шаг 16**

**Задание 16:**

Определить вычеркнутое слово в следующей последовательности:

«Санкт-Петербург, Москва, Владивосток, Екатеринбург, Краснодар, Сочи – города России».

Известно, что каждый символ кодируется **8 битами.** При вычеркивании слова удаляется пробел и запятая. После вычеркивания слова в данной кодировке новый размер последовательности оказался на **12 байтов** меньше, чем размер исходного предложения.

#### ****Решение:****

1. Длина слова = 12 - 2 = 10 символов
2. Анализ:
   * «Санкт-Петербург» – 15
   * «Москва» – 6
   * «Владивосток» – 10
   * «Екатеринбург» – 12
   * «Краснодар» – 9
   * «Сочи» – 4

**Ответ:** Владивосток