**1.**В одной из кодировок Unicode каждый символ кодируется 16 битами. Вова написал текст (в нём нет лишних пробелов):

«Ёж, лев, слон, олень, тюлень, носорог, крокодил, аллигатор  — дикие животные».

Ученик вычеркнул из списка название одного из животных. Заодно он вычеркнул ставшие лишними запятые и пробелы  — два пробела не должны идти подряд.

При этом размер нового предложения в данной кодировке оказался на 16 байт меньше, чем размер исходного предложения. Напишите в ответе вычеркнутое название животного.

**2.**Напишите наименьшее целое число *x*, для которого истинно высказывание:

**НЕ** (*X* <= 8)**И** **НЕ** (*X* >= 15)**И** (*X* чётное).

**3.** У исполнителя Альфа две команды, которым присвоены номера:

**1. прибавь 1;**

**2. умножь на b**

(*b*  — неизвестное натуральное число; *b* ≥ 2).

Выполняя первую из них, Альфа увеличивает число на экране на 1, а выполняя вторую, умножает это число на *b*. Программа для исполнителя Альфа  — это последовательность номеров команд. Известно, что программа **11211** переводит число **3**  в число **62**. Определите значение *b*.

**4.**Ниже приведена программа, записанная на пяти языках программирования.

|  |  |
| --- | --- |
| **Бейсик** | **Python** |
| **DIM** k, s **AS** **INTEGER**  **INPUT** s  **INPUT** k  **IF** s < 5 OR k < 5 **THEN**  **PRINT** "ДА"  **ELSE**  **PRINT** "НЕТ"  **END** **IF** | s = int(input())  k = int(input())  **if** s < 5 **or** k < 5:  **print**("ДА")  **else**:  **print**("НЕТ") |
| **Паскаль** | **Алгоритмический язык** |
| **var** s, k: **integer**;  **begin**  **readln**(s);  **readln**(k);  **if** (s < 5) **or** (k < 5)  **then** **writeln** ('ДА')  **else** **writeln** ('НЕТ')  **end**. | **алг**  **нач**  **цел** s, k  **ввод** s  **ввод** k  **если** s < 5 или k < 5  **то** **вывод** "ДА"  **иначе** **вывод** "НЕТ"  **все**  **кон** |
| **С++** | |
| #include <iostream>  using namespace std;  **int** main() **{**  **int** s, k;      cin >> s;      cin >> k;  **if** (s < 5 || k < 5)          cout << "ДА";  **else**           cout << "НЕТ";  **return** 0;  **}** | |

 Было проведено 9 запусков программы, при которых в качестве значений переменных *s* и *k* вводились следующие пары чисел:

(2, 2); (5, 9); (7, −12); (5, 5); (2, 12); (—10, —13); (—11, 11); (1, 4); (2, 6).

Сколько было запусков, при которых программа напечатала «ДА»?

**5.**В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для логической операции «И» - символ «&». В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

|  |  |
| --- | --- |
| **Запрос** | **Найдено страниц (в тысячах)** |
| Руда | 2200 |
| Уголь | 1300 |
| Руда & Уголь | 200 |

Какое количество страниц (в тысячах) будет найдено по запросу *Руда | Уголь*?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

**6.** Среди приведённых ниже трёх чисел, записанных в десятичной системе счисления, найдите число, в двоичной записи которого наименьшее количество единиц. В ответе запишите количество единиц в двоичной записи этого числа.

5910, 7110, 8110.

**7.**Создайте в текстовом редакторе документ и напишите в нём следующий текст, точно воспроизведя всё оформление текста, имеющееся в образце.

Данный текст должен быть набран шрифтом размером 14 пунктов обычного начертания. Отступ первой строки первого абзаца основного текста  — 1 см. Расстояние между строками текста не менее одинарного, но не более полуторного междустрочного интервала.

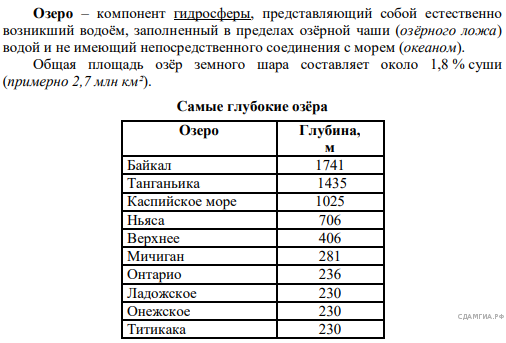
Основной текст выровнен по ширине; в ячейках первого столбца таблицы применено выравнивание по левому краю, в ячейках второго и третьего столбцов  — по центру. В основном тексте и таблице есть слова, выделенные полужирным, курсивным шрифтом и подчёркиванием. Ширина таблицы меньше ширины основного текста. Таблица выровнена на странице

по центру горизонтали.

При этом допустимо, чтобы ширина Вашего текста отличалась от ширины текста в примере, поскольку ширина текста зависит от размеров страницы и полей. В этом случае разбиение текста на строки должно соответствовать стандартной ширине абзаца.

Интервал между текстом и таблицей не менее 12 пунктов, но не более 24 пунктов.

Текст сохраните в файле, имя которого Вам сообщат организаторы. Файл ответа необходимо сохранить в одном из следующих форматов: \*.odt, или \*.doc, или \*.docx.



**8.** Результаты сдачи выпускных экзаменов по алгебре, русскому языку, физике и информатике учащимися 9 класса некоторого города были занесены в электронную таблицу. На рисунке приведены первые строки получившейся таблицы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **А** | **B** | **C** | **D** | **E** | **F** |
| 1 | Фамилия | Имя | Алгебра | Русский | Физика | Информатике |
| 2 | Абапольников | Роман | 4 | 3 | 5 | 3 |
| 3 | Абрамов | Кирилл | 2 | 3 | 3 | 4 |
| 4 | Авдонин | Николай | 4 | 3 | 4 | 3 |

В столбце A электронной таблицы записана фамилия учащегося, в столбце B  — имя учащегося, в столбцах C, D, E и F  — оценки учащегося по алгебре, русскому языку, физике и информатике. Оценки могут принимать значения от 2 до 5. Всего в электронную таблицу были занесены результаты 1000 учащихся.

**Выполните задание**

Откройте файл с данной электронной таблицей. На основании данных, содержащихся в этой таблице, ответьте на два вопроса и постройте диаграмму.

1.  Какое количество учащихся получило удовлетворительные оценки (то есть оценки выше 2) на всех экзаменах? Ответ на этот вопрос запишите в ячейку I2 таблицы.

2.  Для группы учащихся, которые получили удовлетворительные оценки на всех экзаменах, посчитайте средний балл, полученный ими на экзамене **по физике**. Ответ на этот вопрос запишите в ячейку I3 таблицы с точностью не менее двух знаков после запятой.

3.  Постройте круговую диаграмму, отображающую соотношение оценок по информатике «3», «4» и «5». Левый верхний угол диаграммы разместите вблизи ячейки G6.

[task 14.xls](https://inf-oge.sdamgia.ru/get_file?id=20811&png=1)

**\* - Задание на выбор**

**9\*.**Исполнитель Робот умеет перемещаться по лабиринту, начерченному на плоскости, разбитой на клетки. Между соседними (по сторонам) клетками может стоять стена, через которую Робот пройти не может.

У Робота есть девять команд. Четыре команды  — это команды-приказы:

**вверх вниз влево вправо**

При выполнении любой из этих команд Робот перемещается на одну клетку соответственно: вверх ↑ вниз ↓, влево ← , вправо →. Если Робот получит команду передвижения сквозь стену, то он разрушится.

Также у Робота есть команда **закрасить**, при которой закрашивается клетка, в которой Робот находится в настоящий момент.

Ещё четыре команды  —  это команды проверки условий. Эти команды проверяют, свободен ли путь для Робота в каждом из четырёх возможных направлений:

**сверху свободно  снизу свободно  слева свободно  справа свободно**

Эти команды можно использовать вместе с условием **«если»**, имеющим следующий вид:

**если** *условие* **то**

*последовательность команд*

**все**

 Здесь *условие*  — одна из команд проверки условия. *Последовательность команд*  — это одна или несколько любых команд-приказов. Например, для передвижения на одну клетку вправо, если справа нет стенки, и закрашивания клетки можно использовать такой алгоритм:

**если справа свободно то**

**вправо**

**закрасить**

**все**

 В одном условии можно использовать несколько команд проверки условий, применяя логические связки **и, или, не,** например:

**если (справа свободно) и (не снизу свободно) то**

**вправо**

**все**

 Для повторения последовательности команд можно использовать цикл **«пока»**, имеющий следующий вид:

**нц пока** *условие*

*последовательность команд*

**кц**

Например, для движения вправо, пока это возможно, можно использовать следующий алгоритм:

**нц пока справа свободно**

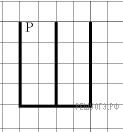
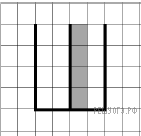
**вправо**

**кц**

**Выполните задание.**

 На бесконечном поле имеется стена, длины отрезков стены неизвестны. Стена состоит из одного горизонтального и трёх равных вертикальных отрезков (отрезки стены расположены буквой «Ш»). Все отрезки неизвестной длины. Робот находится в клетке, расположенной непосредственно справа от верхнего конца левого вертикального отрезка. На рисунке указан один из возможных способов расположения стен и Робота (Робот обозначен буквой «Р») .

 Напишите для Робота алгоритм, закрашивающий все клетки, расположенные непосредственно справа от второго вертикального отрезка. Робот должен закрасить только клетки, удовлетворяющие данному условию. Например, для приведённого выше рисунка Робот должен закрасить следующие клетки (см. рис.).

Конечное расположение Робота может быть произвольным. При исполнении алгоритма Робот не должен разрушиться. Алгоритм должен решать задачу для произвольного размера поля и любого допустимого расположения стен.

 Алгоритм может быть выполнен в среде формального исполнителя или записан в текстовом редакторе.

**10\*.** Напишите программу, которая в последовательности натуральных чисел определяет сумму всех чисел, кратных 6 и оканчивающихся на 4. Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность). Количество чисел не превышает 100. Введённые числа не превышают 300. Программа должна вывести одно число: сумму всех чисел, кратных 6 и оканчивающихся на 4.

**Пример работы программы:**

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| 14  24  36  84  66  0 | 108 |

**Ключ**

|  |  |
| --- | --- |
| **№ п/п** | **Ответ** |
| 1 | тюлень |
| 2 | 10 |
| 3 | 12 |
| 4 | 7 |
| 5 | 3300 |
| 6 | 3 |
| 8 | 1) 795; 2) 3,75. |