

Примеры заданий вступительного испытания по информатике для поступающих в 10 математико-информационный класс

Работа состоит из двух этапов: письменного и практического.

Письменный этап включает задания, выполняемые без использования компьютера. К этим заданиям вы должны самостоятельно сформулировать и записать краткий ответ в указанной форме в отведённом для этого месте.

Практический этап включает задания, для выполнения которых вам может потребоваться компьютер. Можно использовать все приложения (текстовый редактор Блокнот, графический редактор Paint, текстовый процессор MS Word, средство работы с электронными таблицами MS Excel) и среды программирования (построитель блок-схем АЛГИС, среда исполнителя Кумир, PascalABC для Pascal, CodeBlocks для C++, MS Visual Studio 17 для C++ и C#, PyCharm для Python и JavaScript, Eclipse для Java), установленные на предоставленном компьютере. К этим заданиям необходимо написать полное решение в произвольной форме – или в виде файла, или на бумаге. Ответ должен быть обязательно записан на бумаге, а в случае решения в файле на бумаге должно быть отмечено имя этого файла.

Письменный этап

В качестве примеров заданий письменного этапа могут рассматриваться задания вступительных испытаний 2015 – 2020 г.г.

Практический этап

1. В электронную таблицу занесли данные о тестировании учеников. Ниже приведены первые пять строк таблицы:

	A	B	C	D
1	округ	фамилия	предмет	балл
2	C	Ученик 1	обществознание	246
3	B	Ученик 2	немецкий язык	530
4	Ю	Ученик 3	русский язык	576
5	CB	Ученик 4	обществознание	304

В столбце A записан округ, в котором учится ученик; в столбце B — фамилия; в столбце C — любимый предмет; в столбце D — тестовый балл. Всего в электронную таблицу были занесены данные по 1000 ученикам.

Выполните задание.

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

1. Сколько учеников в Северо-Западном округе (C3) выбрали в качестве любимого предмета русский язык? Ответ на этот вопрос запишите в ячейку H2 таблицы.
2. Каков средний тестовый балл у учеников Западного округа (3)? Ответ на этот вопрос запишите в ячейку H3 таблицы с точностью не менее двух знаков после запятой.

3. Постройте круговую диаграмму, отображающую соотношение числа участников, сдающих немецкий язык, русский язык и обществознание. Левый верхний угол диаграммы разместите вблизи ячейки G6.

Решение.

1. Запишем в ячейку E2 следующую формулу =ЕСЛИ(A2="СЗ";С2;0) и скопируем ее в диапазон E3:E1001. В ячейку столбца E будет записываться название предмета, если ученик из Северо-Западного округа и «0», если это не так. Применив операцию =ЕСЛИ(E2="русский язык";1;0), получим столбец (F) с единицами и нулями. Далее используем операцию =СУММ(F2:F1001). Получим количество учеников, которые считают своим любимым предметом русский язык.

2. Для ответа на второй вопрос используем операцию «ЕСЛИ». Запишем в ячейку G2 следующее выражение: =ЕСЛИ(A2="З";D2;0), в результате применения данной операции к диапазону ячеек G2:G1001, получим столбец, в котором записаны баллы только учеников Западного округа. Сложив значения в ячейках, получим сумму баллов учеников. Найдём количество учеников Западного округа с помощью команды =СЧЁТЕСЛИ(A2:A1001;"З"). Разделив сумму баллов на количество учеников, получим искомый средний балл.

3. В ячейку J2 вставим формулу =СЧЁТЕСЛИ(C2:C1001; "немецкий язык"), в ячейку J3 вставим формулу =СЧЁТЕСЛИ(C2:C1001; "русский язык"), в ячейку J4 вставим формулу =СЧЁТЕСЛИ(C2:C1001; "обществознание"). Теперь построим по полученным значениям круговую диаграмму, подпишем сектора.

2. Найдите все натуральные числа, меньшие 110, которые имеют ровно три различных простых делителя. Например, число 120 подходило бы, так как оно делится ровно на три простых числа – на 2, на 3 и на 5, но оно больше 110. Запишите в ответе все искомые числа в порядке возрастания через запятую.

Решение.

1 способ. Можно просто перебирать все числа подряд от 1 до 109, находя для каждого числа все его простые делители и их количество. Числа, у которых простых делителей окажется ровно 3, записываем в ответ. Получится последовательность 30, 42, 60, 66, 70, 78, 84, 90, 102, 105.

2 способ. Выпишем простые числа от 2 до 20: 2, 3, 5, 7, 11, 13, 17, 19. Рассмотрим произведения этих чисел из трех множителей, учитывая, что они могут повторяться

Делители	Полученные числа
2, 3, 5	$2 \times 3 \times 5 = 30$; $2 \times 2 \times 3 \times 5 = 60$; $2 \times 3 \times 3 \times 5 = 90$; $2 \times 2 \times 2 \times 3 \times 5 > 110$;
2, 3, 7	$2 \times 3 \times 7 = 42$; $2 \times 2 \times 3 \times 7 = 84$; $2 \times 3 \times 3 \times 7 > 110$;
2, 3, 11	$2 \times 3 \times 11 = 66$; $2 \times 2 \times 3 \times 11 > 110$;
2, 3, 13	$2 \times 3 \times 13 = 78$; $2 \times 2 \times 3 \times 13 > 110$;
2, 3, 17	$2 \times 3 \times 17 = 102$; $2 \times 2 \times 3 \times 17 > 110$;
2, 3, 19	$2 \times 3 \times 19 > 110$;
2, 5, 7	$2 \times 5 \times 7 = 70$; $2 \times 2 \times 5 \times 7 > 110$;
2, 5, 11	$2 \times 5 \times 11 = 110$ (не подходит, так как 110 не меньше, чем 110);
2, 7, 11	$2 \times 7 \times 11 > 110$;
3, 5, 7	$3 \times 5 \times 7 = 105$; $3 \times 3 \times 5 \times 7 > 110$;
3, 5, 11	$3 \times 5 \times 11 > 110$;

Запишем все полученные числа в порядке возрастания через запятую.

Ответ: 30, 42, 60, 66, 70, 78, 84, 90, 102, 105.

3. Определите, существует ли система счисления, в которой выполняются равенства:

$$3 + 4 = 7,$$

$$3 * 4 = 13,$$

$$39 + 29 = 70.$$

Ответ обосновать.

Решение. Такой системы счисления не существует. Предположим, что такая p-ичная система счисления существует, тогда

$$3p^0 + 4p^0 = 7p^0$$

$$3p^0 * 4p^0 = p + 3p^0$$

$$3p + 9 + 2p + 9 = 7p$$

Решением этой системы является $p = 9$. Но в системе счисления с основанием 9 записать числа 29 и 39 невозможно. Следовательно, такой системы счисления не существует.

Ответ. Не существует.

4. Упростите логическое выражение или укажите его результат (при его однозначности). Результат упрощения может содержать только операции инверсии, конъюнкции и дизъюнкции.
(не А или В) и не В и не (не А и не В)

(привести полное решение).

Решение. (не А или В) и не В и не (не А и не В) = (закон де Моргана) = (не А или В) и не В и (А или В) = (распределительный закон) = (не А и не В или В и не В) и (А или В) = (закон исключения третьего) не А и не В и (А или В) = (распределительный закон) не А и не В и А или не А и не В и В = 0 или 0 = 0.

Ответ. 0.

Примечание. Можно также решать эту задачу построением таблицы истинности.

5. Палиндромом называется слово, которое одинаково читается как слева направо, так и справа налево, например, в английском языке такими словами являются «radar» и «racesar». Света изучает английский язык и решила принять участие в дистанционном конкурсе знатоков английского языка. Но, когда она писала ответ на задание «найдите самое длинное слово, которое является палиндромом», ошиблась и нажала на клавиатуре одну лишнюю клавишу. Определите, какую букву нужно удалить в набранном Светой слове, чтобы это слово стало палиндромом. Программа получает на вход строку из строчных английских букв, содержащую не менее 2 и не более 100 000 символов. Программа должна вывести единственное число – номер буквы в строке, при удалении которой слово становится палиндромом. Если при удалении любой буквы слово не станет палиндромом, программа должна вывести число 0.

Примеры входных и выходных данных

Ввод	Вывод
raceszar	6
car	0

Решение. Для начала напишем простой перебор всех возможных ответов: переберём все символы строки, удалим их, если после удаления символа получится палиндром, выведем номер этого символа. Если не нашлось такого символа, выведем 0. Пример решения.

```
s = input()
for i in range(len(s)):
    if s[:i] + s[i + 1:] == (s[:i] + s[i + 1:])[::-1]:
        print(i + 1)
        break
else:
    print(0)
```

В этом решении используется ряд удобств языка Python, поэтому объясним подробнее. `s[:i]` – это срез строки: подстрока, состоящая из символов до `i`-го, не включая `i`-й символ. `s[i + 1:]` – это срез от символа с индексом `i + 1` до конца строки. `s[:i] + s[i + 1:]` – это конкатенация двух срезов, то есть исходная строка без символа с индексом `i`. `[::-1]` – это срез из символов строки, взятых с шагом `-1`, то есть строка, развёрнутая в обратном порядке, то есть в инструкции `if` проверяется, верно ли, что после выбрасывания символа с индексом `i` получается палиндром. Программа выводит `i+1`, потому что в языке Python индексация элементов строки начинается с `0`, а нужно вывести ответ в нумерации символов с `1`. Инструкция `else` относится к циклу `for`, она выполняется после завершения цикла. Если цикл был прерван по инструкции `break`, то `else` не выполняется, поэтому число `0` будет напечатано, только если ответ не был найден ранее. Поскольку проверка на палиндромность осуществляется за время, пропорциональное длине строки (нужно развернуть строку и сравнить две строки), то такое решение будет иметь сложность $O(n^2)$, где n – длина строки. Приведём решение, имеющее сложность $O(n)$. Сравним первый и последний символ. Если они различаются, то, значит, один из них должен быть удалён. Если они равны, то сравним второй и предпоследний символ, если они равны, то перейдём к следующим от концов строки символам и т.д. Нам необходимо найти такие два символа, отстоящие на равное расстояние от концов строки, которые различаются. Один из этих символов должен быть удалён. Проверим, что после удаления одного из этих символов строка становится палиндромом, то есть ответ найден. Если же при удалении каждого из этих двух символов строка не становится палиндромом, то задача не имеет решения и нужно вывести `0`. Наконец, если такие символы не были найдены, то строка уже является палиндромом. В этом случае

нужно удалить средний символ в строке (в случае строки чётной длины – один из двух средних символов). Пример решения.

```
s = input()
i = 0
j = len(s) - 1
while i < j and s[i] == s[j]:
    i += 1    # Поиск двух различных символов i и j,
    j -= 1    # равноудалённых от концов строки
if i >= j:
    print(i + 1) # строка – уже палиндром, выводим центральный символ
elif s[:i] + s[i + 1:] == (s[:i] + s[i + 1:])[::-1]:
    print(i + 1) # можно удалить символ i
elif s[:j] + s[j + 1:] == (s[:j] + s[j + 1:])[::-1]:
    print(j + 1) # можно удалить символ j
else:
    print(0) # решения нет
```