

СПРАВОЧНЫЕ МАТЕРИАЛЫ

ОГЭ 2027 • ИНФОРМАТИКА

Краткий конспект по всем ключевым темам ОГЭ: информация и кодирование, логика, сети, системы счисления, файлы, таблицы, Робот и программирование.

Формат ОГЭ-2027

По проекту ФИПИ структура и содержание КИМ не изменены: **16 заданий**, максимальный результат - **21 первичный балл**, время - **150 минут**. Задания 1-10 - часть 1; задания 11-16 выполняются с использованием компьютера. Для 13-16 результатом является отдельный файл. В задании 13 выбирают **13.1 (презентация)** или **13.2 (текстовый документ)**.

1. ИНФОРМАЦИЯ И ОБЪЁМ ДАННЫХ - ЗАДАНИЕ 1

Единица	Соотношение	Запомнить
бит	минимальная единица информации	0 или 1
байт	1 байт = 8 бит	В / Б
Кбайт	1 Кбайт = 1024 байт	2^{10} байт
Мбайт	1 Мбайт = 1024 Кбайт	2^{20} байт
Гбайт	1 Гбайт = 1024 Мбайт	2^{30} байт

Текст из K символов: $I = K \cdot i$, где I - объём сообщения, i - число бит на символ. Если алфавит содержит N символов и используется равномерный двоичный код минимальной длины, то $i = \lceil \log_2 N \rceil$.

Мощность алфавита N	2	4	8	16	32	64	128	256
Минимум бит/символ	1	2	3	4	5	6	7	8

- Если в условии прямо сказано «16 бит на символ», не вычисляй минимум: бери **$i = 16$ бит = 2 байта**.
- После изменения текста: новый объём = новое количество символов × бит на символ. Пробел, знак препинания и перевод строки учитываются только если это следует из условия.

Быстрая проверка единиц

Сначала приведи всё к одной единице. Биты -> байты: дели на 8. Байты -> Кбайты: дели на 1024. При переходе в меньшую единицу - умножай.

2. КОДИРОВАНИЕ И ДЕКОДИРОВАНИЕ - ЗАДАНИЕ 2

Равномерный код - все кодовые слова одной длины: последовательность просто делится на блоки.

Неравномерный код - длины различаются; разбиение нужно подбирать так, чтобы вся строка была разобрана без остатка.

Условие Фано: ни одно кодовое слово не является началом другого. Тогда декодирование слева направо однозначно. Если условие Фано не выполнено, проверяй все допустимые разбиения.

Шаг	Что делать
1	Выпиши таблицу «символ -> код» и отметь самые короткие коды.
2	Читай двоичную строку слева направо; как только найден полный код, фиксируй символ.
3	Если есть несколько вариантов продолжения - сделай ветвление и отбрось путь, который упирается в невозможный остаток.
4	В конце вся последовательность должна быть использована; лишних битов быть не должно.

3. ЛОГИКА - ЗАДАНИЕ 3

Операция	Запись	Когда истинно	Python
НЕ А	$\neg A$	когда А ложно	not A
А И В	$A \wedge B$	истинны оба	A and B
А ИЛИ В	$A \vee B$	истинно хотя бы одно	A or B
Равно / не равно	$= / \neq$	сравнение значений	$== / !=$

Приоритет: скобки $\rightarrow$ НЕ $\rightarrow$ И $\rightarrow$ ИЛИ. Если сомневаешься, расставь скобки вручную.

Полезные преобразования: $\text{НЕ}(x > a) = x \leq a$; $\text{НЕ}(x \geq a) = x < a$; $\text{НЕ}(A \text{ И } B) = \text{НЕ } A \text{ ИЛИ } \text{НЕ } B$; $\text{НЕ}(A \text{ ИЛИ } B) = \text{НЕ } A \text{ И } \text{НЕ } B$.

Фраза	Логическая запись
x принадлежит [a; b]	$x \geq a \text{ И } x \leq b$
x вне [a; b]	$x < a \text{ ИЛИ } x > b$
x кратно k	$x \bmod k = 0$
x нечётное	$x \bmod 2 \neq 0$
последняя цифра d	$x \bmod 10 = d$

4. ГРАФЫ И МОДЕЛИ - ЗАДАНИЯ 4 И 9

Граф: вершины - объекты, рёбра - связи. Взвешенный граф имеет числа на рёбрах (расстояние, время, стоимость). **Длина пути** - сумма весов всех пройденных рёбер.

- **Таблица смежности:** ненулевое число обычно означает наличие дороги/ребра и его вес; пустая клетка или «-» - связи нет. Для неориентированного графа таблица симметрична относительно главной диагонали.
- **Кратчайший путь:** перебирай только допустимые маршруты; как только частичная сумма уже больше лучшего найденного результата, этот маршрут можно не продолжать.
- **Количество путей в ориентированной схеме:** поставь у старта 1; у каждой следующей вершины запиши сумму чисел во всех вершинах, из которых в неё идут стрелки. Работает для схем без циклического возврата.

Не смешивай два типа задач

В задачах на кратчайший путь складывают **веса рёбер**. В задачах на количество путей складывают **числа способов попасть в вершину**.

5. ИСПОЛНИТЕЛЬ И ПРОСТЫЕ АЛГОРИТМЫ - ЗАДАНИЕ 5

Команды вида «прибавь а», «умножь на b», «вычти с» выполняются строго по порядку. Если программа задана последовательностью номеров команд, замени каждый номер соответствующей командой и веди промежуточное значение.

- Если нужно получить целевое число, часто быстрее идти **назад**: обратная к «+а» - «-а», обратная к «x b» - «÷b» (только если делится нацело).
- Если нужно посчитать количество программ, используй динамику: для каждого числа считай, сколькими способами в него можно прийти из предыдущих допустимых состояний.

6. ФРАГМЕНТ ПРОГРАММЫ - ЗАДАНИЕ 6

Главная техника - **трассировка**. Подставляй входные значения, выполняй присваивания сверху вниз и отдельно вычисляй каждое условие. После присваивания старое значение переменной теряется.

Операция	Python	Смысл
остаток	$a \% b$	остаток от деления a на b
целая часть	$a // b$	целочисленное деление
степень	$a ** b$	a в степени b
присваивание	$x = x + 1$	новое x равно старому x + 1
сравнение	$x == 5$	проверка равенства, значение x не меняется

7. АДРЕСАЦИЯ В ИНТЕРНЕТЕ - ЗАДАНИЕ 7

Типовая структура URL: **протокол://сервер/путь/файл.расширение**. Пример:
https://site.ru/folder/page.html.

Часть	Пример	Важно
протокол	http, https, ftp	после протокола обычно ://
сервер / домен	school.ru	точки разделяют части доменного имени
путь	docs/2027/	папки разделяются /
имя файла	task.pdf	точка отделяет расширение

Сборка адреса из фрагментов

Сначала найди протокол и «://», затем домен, после него путь через «/», в конце имя файла и расширение. Не ставь «/» внутри имени домена и не путай точку домена с точкой расширения.

8. ПОИСКОВЫЕ ЗАПРОСЫ И МНОЖЕСТВА - ЗАДАНИЕ 8

Результаты поискового запроса удобно считать множествами страниц. **И / &** - пересечение, **ИЛИ / |** - объединение. Чем больше условий соединено «И», тем обычно меньше результатов; «ИЛИ» расширяет выдачу.

Формула	Смысл
$ A \cup B = A + B - A \cap B $	объединение двух множеств
$ A \cap B = A + B - A \cup B $	пересечение двух множеств
$ A \cup B \cup C = A + B + C - A \cap B - A \cap C - B \cap C + A \cap B \cap C $	объединение трёх множеств

- Если дана диаграмма Эйлера, сначала подпиши области: только А, только В, А∩В без С и т.д. Тогда любой запрос превращается в сумму нужных областей.

9. СИСТЕМЫ СЧИСЛЕНИЯ - ЗАДАНИЕ 10

Из системы с основанием b в десятичную: умножай каждую цифру на соответствующую степень b и складывай. Например, $231_4 = 2 \cdot 4^2 + 3 \cdot 4 + 1$.

Из десятичной в систему b: дели число на b, записывай остатки и читай их снизу вверх. Для оснований больше 10: А=10, В=11, С=12, D=13, E=14, F=15.

Основание	Цифры	Полезный переход
2	0-1	1 восьмеричная цифра = 3 двоичных бита
8	0-7	разбивай двоичную запись справа на тройки
16	0-9, A-F	1 шестнадцатеричная цифра = 4 двоичных бита

- Последняя цифра записи числа в системе b равна $x \bmod b$. Это особенно полезно в задании 16.
- k-значное положительное число в системе b удовлетворяет: $b^{(k-1)} \leq x < b^k$.

10. ФАЙЛЫ, ПАПКИ И МАСКИ - ЗАДАНИЯ 11-12

Задания выполняются на компьютере с предоставленным каталогом. Перед поиском проверь, **в какой папке искать**, нужен ли поиск по имени файла или по содержимому, и учитываются ли вложенные папки.

Символ маски	Что означает	Пример
*	любая последовательность символов, в том числе пустая	a*.txt -> a.txt, abc.txt
?	ровно один любой символ	a?.txt -> ab.txt, a7.txt
.	обычный символ точки	*.pdf -> все PDF-файлы

- Для каждой маски проверяй отдельно **имя** и **расширение**. Символ «?» нельзя пропустить: он обязан соответствовать ровно одному символу.
- При поиске текста внутри файлов используй точное слово/фразу из условия и затем перепроверь найденные файлы вручную: слово может встречаться в заголовке, таблице или другом контексте.

Информационный объём группы файлов: общий размер = сумма размеров всех подходящих файлов. Если файлы одинакового размера: $V = n \cdot v$. Переводи единицы только после подсчёта.

Практический порядок

- 1) Открой нужную папку. 2) Включи отображение расширений и при необходимости сортировку. 3) Применяй условие/маску. 4) Пересчитай результат и проверь пограничные файлы. 5) Для объёма не путай размер одного файла и суммарный размер.

11. ЗАДАНИЕ 13: ПРЕЗЕНТАЦИЯ ИЛИ ТЕКСТОВЫЙ ДОКУМЕНТ

13.1 - презентация	13.2 - текстовый документ
Сделай ровно столько слайдов, сколько указано (в типовом задании - 3).	Точно воспроизведи текст, абзацы, выделения и таблицу по образцу.
Соблюдай макет: заголовки, текстовые блоки, изображения на заданных местах.	Размер шрифта, выравнивание, отступ первой строки и интервалы задавай настройками абзаца.
Единый тип шрифта и читаемые размеры; текст не перекрывает изображения.	Не имитируй отступы множеством пробелов и не разбивай строки вручную Enter/Shift+Enter без необходимости.
Изображения не растягивай с искажением пропорций и не обрезай важные части.	Таблица должна иметь правильное число строк/столбцов; проверяй верхние индексы и специальные символы.
Сохрани в указанном формате и под именем, которое сообщат организаторы.	Сохрани в указанном формате и под именем, которое сообщат организаторы.

Главное правило задания 13

Не «улучшай дизайн» от себя. Эксперт оценивает соответствие условию, макету и контролируемым элементам. Сначала точность, потом аккуратность.

12. ЭЛЕКТРОННЫЕ ТАБЛИЦЫ - ЗАДАНИЕ 14

Задание обычно состоит из нескольких подпунктов: посчитать показатель по большой таблице и записать ответы в заданные ячейки, а затем построить диаграмму/график по указанным данным. Итоговый файл нужно сохранить.

Что нужно	Формула / приём
Сумма	SUM / СУММ
Среднее	AVERAGE / СРЗНАЧ
Количество чисел	COUNT / СЧЁТ
Количество по условию	COUNTIF / СЧЁТЕСЛИ
Несколько условий	COUNTIFS / СЧЁТЕСЛИМН или вспомогательный столбец
Сумма по условию	SUMIF / СУММЕСЛИ
Среднее по условию	AVERAGEIF / СРЗНАЧЕСЛИ

Критерии в формулах: текст - в кавычках («Заречный»), сравнение - тоже в кавычках («>140»). Разделитель аргументов зависит от программы и локали: чаще ; в русской локали и , в английской.

Ссылка	Что происходит при копировании
A2	относительная - меняются строка и столбец
\$A\$2	абсолютная - не меняется
\$A2	фиксирован столбец A
A\$2	фиксирована строка 2

- Для условия «сумма двух предметов больше 140» безопасно создать вспомогательный столбец **=B2+C2**, протянуть формулу и считать подходящие строки.
- Перед построением диаграммы выделяй только нужный диапазон. Проверь тип диаграммы, подписи категорий, легенду и наличие заголовка, если это требуется условием.
- Не удаляй исходные данные и не переименовывай листы без необходимости. Сохраняй после каждого завершённого подпункта.

Проверка задания 14

Открой ячейку с ответом: там должно быть число или формула, дающая нужное число. Затем кликни по диаграмме и проверь, что она построена именно по требуемым данным, а не по соседним столбцам.

13. ПОЛЕЗНЫЕ ПРИЁМЫ ДЛЯ УСЛОВИЙ В ТАБЛИЦАХ

Условие	Как проверить
$x > 140$	$x > 140$
от 50 до 80 включительно	$x \geq 50$ И $x \leq 80$
район = Заречный	точное совпадение текста
начинается с А	маска А* или функция LEFT/ЛЕВСИМВ при необходимости
не равно значению	<> в электронных таблицах

14. ИСПОЛНИТЕЛЬ «РОБОТ» - ЗАДАНИЕ 15

Цель - написать алгоритм, который работает **при всех допустимых исходных данных**, не разбивает Робота и завершается. Нельзя ориентироваться только на размеры стены на рисунке, если в условии сказано, что они неизвестны.

Группа	Команды / условия
Движение	вверх, вниз, влево, вправо
Действие	закрасить
Проверки	сверху свободно, снизу свободно, слева свободно, справа свободно
Логика	и, или, не
Цикл	нц пока <условие> ... кц
Ветвление	если <условие> то ... все

Шаблон: дойти до стены

```
нц пока сверху свободно
  вверх
кц
```

Шаблон: идти вдоль стены и закрашивать

```
нц пока не снизу свободно
  закрасить
  вправо
кц
закрасить | если последняя клетка тоже нужна
```

- Переводи условие цикла словами: «иду вправо вдоль стены снизу» -> справа путь может быть свободен, но **снизу должна оставаться стена**, то есть «не снизу свободно».
- Если старт Робота произвольный, сначала перемести его к однозначному ориентиру (краю стены/концу коридора), не закрашивая лишних клеток.
- Проверь минимум три ситуации: самая короткая допустимая стена, длинная стена, расположение с проходами/углами, если они разрешены условием.

Шаблон: закрасить весь узкий вертикальный коридор

```
нц пока сверху свободно
  вверх
кц
нц пока снизу свободно
  закрасить
  вниз
кц
закрасить
```

Частая ошибка	Как исправить
Посчитано число шагов по рисунку	Заменить константу циклом с условием свободно/стена.
Робот пытается пройти сквозь стену	Перед движением проверить соответствующее условие или изменить условие цикла.
Не закрашена последняя клетка	Понять, на какой клетке цикл завершился; при необходимости добавить закрасить после цикла.
Алгоритм работает только для одного старта	Сначала привести Робота к однозначному ориентиру.

Критерий максимума

Для полного балла алгоритм должен правильно работать при всех допустимых исходных данных. Проверять не рисунок, а словесные ограничения задачи.

15. ПРОГРАММИРОВАНИЕ - ЗАДАНИЕ 16

Нужно написать и сохранить программу на универсальном языке программирования. Типовая задача: обработать последовательность чисел по условию и вывести количество, сумму, максимум/минимум, среднее или YES/NO.

Условие о числе	Python
кратно k	<code>x % k == 0</code>
не кратно k	<code>x % k != 0</code>
чётное / нечётное	<code>x % 2 == 0 / x % 2 != 0</code>
оканчивается цифрой d	<code>x % 10 == d</code>
в системе b оканчивается цифрой d	<code>x % b == d</code>
k-значное в системе b	<code>b**(k-1) <= x < b**k</code>

Шаблон А: сначала дано количество N

```
n = int(input())
count = 0
total = 0
for _ in range(n):
    x = int(input())
    if УСЛОВИЕ:
        count += 1
        total += x
print(count) # или total
```

Шаблон В: последовательность заканчивается нулём

```
count = 0
while True:
    x = int(input())
    if x == 0:
        break
    if УСЛОВИЕ:
        count += 1
print(count)
```

Среднее: храни сумму *total* и количество *count*, затем `total / count`. Если по условию подходящих элементов может не быть, перед делением проверь `count > 0` и выведи требуемое сообщение (например, NO).

Максимум/минимум: безопасный способ - при первом подходящем числе присвоить ему `max/min`, затем обновлять. Не инициализируй максимум нулём, если возможны отрицательные числа.

Финальная проверка программы

- 1) Перепроверь формат ввода: N или завершающий 0. 2) Ноль-признак не входит в последовательность. 3) Выполни пример из условия. 4) Проверь случай без подходящих чисел, если он допускается. 5) Сохрани финальную версию файла после тестирования.

16. ЧТО ПРОВЕРИТЬ ПЕРЕД СДАЧЕЙ

- Ответы 1-12 перенесены без лишних символов и в нужном порядке.
- Файлы 13-16 сохранены **в нужном каталоге**, с нужными именами и расширениями.
- Задание 13: выбран только один вариант - 13.1 или 13.2 - и выполнены требования именно своего варианта.
- Задание 14: формулы дают нужные числа, диаграмма построена по правильному диапазону.
- Задание 15: алгоритм не зависит от случайного размера рисунка и не разбивает Робота.
- Задание 16: программа проходит пример и хотя бы 2 собственных граничных теста.

Основа: проект демоверсии, спецификации и кодификатора ОГЭ-2027 ФИПИ по информатике; методические материалы ФИПИ по проверке практической части. На дату подготовки документы ОГЭ-2027 опубликованы как проекты.