

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**



**ОЛИМПИАДА ШКОЛЬНИКОВ
«ГРАНИТ НАУКИ»**

ИНФОРМАТИКА

**Методические указания для подготовки к
заключительному туру олимпиады в 2021/2022
учебном году**

Санкт-Петербург
2022



ОГЛАВЛЕНИЕ

Информатика	3
Задания для подготовки к олимпиаде	6
Список рекомендуемой литературы для подготовки	21



Информатика

Характер и уровень сложности олимпиадных задач направлены на достижение целей проведения олимпиады: выявить способных участников, твердо владеющих школьной программой, обладающих логикой и творческим характером мышления, умеющих алгоритмически описать реальные ситуации из различных предметных областей и применить к ним наиболее подходящие методы информатики.

Задания дифференцированы по сложности и требуют различных временных затрат на верное и полное решение. Они охватывают все разделы школьной программы, но носят, в большинстве, комплексный характер, позволяющий варьировать оценки в зависимости от проявленных в решении творческих подходов и продемонстрированных технических навыков. Участники должны самостоятельно разбить задачу на подзадачи, грамотно выполнить решение каждой подзадачи, синтезировать решение всей задачи из решений отдельных подзадач. Успешное выполнение олимпиадной работы не предполагает знаний, выходящих за пределы школьной программы, но требует творческого подхода, логического мышления, умения увидеть и составить правильный и оптимальный план решения, четкого и технически грамотного выполнения каждой части решения.

Очный тур олимпиады проводится только в письменной форме. Каждый участник олимпиады получает билет, содержащий шесть заданий. При выполнении заданий требуется дать теоретическое обоснование и привести все этапы решения задачи:

- при определении результата перевода заданного числа из одной системы счисления в другую следует указать правило перевода в общем виде, а затем привести схему перевода заданного числа в указанную систему счисления;

- при определении значений ячеек электронной таблицы приводятся результаты, формулы, полученные при копировании, даются пояснения о типах ссылок в формулах, как типы ссылок влияют на результат и др.;

- при ответе на вопрос об объемах памяти, выделяемой для кодирования символьной информации, необходимо указать особенности системы кодирования, привести подробные расчеты;

- при ответе на вопрос о поисковых запросах следует привести результаты промежуточных расчетов, полученных после выполнения запросов;



– при ответе на вопрос об определении значения переменной после выполнения программы, алгоритм которой приводится в билете, следует пояснить работу каждого оператора (блока), выписать промежуточные и окончательные результаты;

– необходимо дать развернутый ответ с пояснениями при составлении расширенного алгоритма, пояснить алгоритм графически;

– ответы на задачи, посвященные темам, связанным с сетями, IP-адресами, масками подсети должны сопровождаться публикацией всей информации без сокращений;

– в случае решения задач, связанных с таблицами истинности необходимо описывать все промежуточные результаты;

– при решении задачи на программирование необходимо составить блок-схему или описать алгоритм решения задачи, написать текст программы на любом из языков программирования, указав используемый язык программирования и его версию, допускается использовать только стандартную библиотеку языка;

– алгоритм должен раскрывать решение на каждом шаге. Код программы должен состоять из простых и понятных команд. Запрещено использовать уникальные особенности языка программирования (классы, методы, объекты, попарный обмен между переменными, специальные функции и библиотеки и др.) для реализации решения. Считывание заранее рассчитанных значений не засчитывается за решение задачи. Код должен содержать достаточное для понимания логики работы количество комментариев;

– программы, составленные участниками, тестируются и всесторонне оцениваются с учетом быстродействия, рациональности, минимального использования памяти компьютера и оригинальности предложенных решений.

При подготовке к олимпиаде следует повторить приведенные ниже разделы.

1. Информация и информационные процессы

1.1. Информация и ее кодирование

1.2. Виды информационных процессов

1.3. Процесс передачи информации, источник и приемник информации.

Сигнал, кодирование и декодирование.

1.4. Искажение информации

1.5. Дискретное (цифровое) представление текстовой, графической, звуковой информации и видеoinформации. Единицы измерения



количества информации

1.6. Скорость передачи информации

1.7. Системы, компоненты, состояние и взаимодействие компонентов. Информационное взаимодействие в системе, управление, обратная связь

1.8. Моделирование

1.9. Описание (информационная модель) реального объекта и процесса, соответствие описания объекту и целям описания. Схемы, таблицы, графики, формулы как описания

1.10. Математические модели

1.11. Использование сред имитационного моделирования (виртуальных лабораторий) для проведения компьютерного эксперимента в учебной деятельности

1.12. Системы счисления

1.13. Позиционные системы счисления

1.14. Двоичное представление информации

1.15. Логика и алгоритмы

1.16. Высказывания, логические операции, кванторы, истинность высказывания

1.17. Цепочки (конечные последовательности), деревья, списки, графы, матрицы (массивы), псевдослучайные последовательности

1.18. Индуктивное определение объектов

1.19. Вычислимые функции, полнота формализации понятия вычислимости, универсальная вычислимая функция

1.20. Кодирование с исправлением ошибок

1.21. Сортировка

1.22. Элементы теории алгоритмов

1.23. Формализация понятия алгоритма

1.24. Вычислимость. Эквивалентность алгоритмических моделей

1.25. Построение алгоритмов и практические вычисления

1.26. Языки программирования

1.27. Типы данных

1.28. Основные конструкции языка программирования. Система программирования

1.29. Основные этапы разработки программ. Разбиение задачи на подзадачи

2. Информационная деятельность человека

2.1. Профессиональная информационная деятельность. Информационные



ресурсы

2.2. Экономика информационной сферы

2.3. Информационная этика и право, информационная безопасность



3. Средства ИКТ

- 3.1. Архитектура компьютеров и компьютерных сетей
- 3.2. Программная и аппаратная организация компьютеров и компьютерных систем. Виды программного обеспечения
- 3.3. Операционные системы. Понятие о системном администрировании
- 3.4. Безопасность, гигиена, эргономика, ресурсосбережение, технологические требования при эксплуатации компьютерного рабочего места
- 3.5. Технологии создания и обработки текстовой информации
- 3.6. Понятие о настольных издательских системах. Создание компьютерных публикаций
- 3.7. Использование готовых и создание собственных шаблонов. Использование систем проверки орфографии и грамматики. Тезаурусы. Использование систем двуязычного перевода и электронных словарей
- 3.8. Использование специализированных средств редактирования математических текстов и графического представления математических объектов
- 3.9. Использование систем распознавания текстов
- 3.10. Технология создания и обработки графической и мультимедийной информации
- 3.11. Форматы графических и звуковых объектов
- 3.12. Ввод и обработка графических объектов
- 3.13. Ввод и обработка звуковых объектов
- 3.14. Обработка числовой информации
- 3.15. Математическая обработка статистических данных
- 3.16. Использование динамических (электронных) таблиц для выполнения учебных заданий из различных предметных областей
- 3.17. Использование инструментов решения статистических и расчетно-графических задач
- 3.18. Технологии поиска и хранения информации
- 3.19. Системы управления базами данных. Организация баз данных
- 3.20. Использование инструментов поисковых систем (формирование запросов)
- 3.21. Телекоммуникационные технологии
- 3.22. Специальное программное обеспечение средств телекоммуникационных технологий
- 3.23. Инструменты создания информационных объектов для Интернет

**Задания для подготовки к олимпиаде**

Задача 1. Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч (по своему выбору) один камень или увеличить количество камней в куче в два раза. Например, пусть в одной куче 10 камней, а в другой 7 камней; такую позицию в игре будем обозначать (10, 7). Тогда за один ход можно получить любую из четырёх позиций: (11, 7), (20, 7), (10, 8), (10, 14). Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 73. Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, что в кучах всего будет 73 камня или больше.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. Например, при начальных позициях (6, 34), (7, 33), (9, 32) выигрышная стратегия есть у Пети. Чтобы выиграть, ему достаточно удвоить количество камней во второй куче.

Задание 1. Для каждой из начальных позиций (6, 33), (8, 32) укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

Задание 2. Для каждой из начальных позиций (6, 32), (7, 32), (8, 31) укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

Задание 3. Для начальной позиции (7, 31) укажите, кто из игроков имеет выигрышную стратегию. Опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии. Постройте дерево всех партий, возможных при указанной Вами выигрышной стратегии. Представьте дерево в виде рисунка или таблицы.



Ответ: 1. Ваня первым ходом.

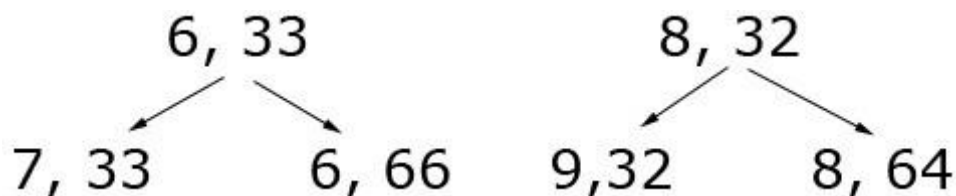
2. Петя вторым ходом.

3. Ваня вторым ходом.

Решение:

Задание 1. Кто из игроков имеет выигрышную стратегию при позициях (6, 33) и (8, 32)?

Игроки могут делать два хода — прибавить к любой из куч один камень, или умножить количество камней в любой из куч на два. Игрок выигрывает, когда суммарное количество камней в куче становится ≥ 73 . Будем считать, что прибавление одного камня в маленькую кучу — самый слабый ход, а увеличение в два раза большой кучи — самый сильный ход. Остальные ходы нам не интересны. Рассмотрим это на графе:



Петя первым ходом не может получить суммарное количество ≥ 73 , зато Ваня может, удвоив количество камней во второй куче после хода Пети:

$$7 + 2 \cdot 33 = 73$$

$$6 + 2 \cdot 66 = 138$$

$$9 + 2 \cdot 32 = 73$$

$$8 + 2 \cdot 64 = 136$$

То есть при позициях (6, 33) и (8, 32) второй игрок (Ваня) выигрывает первым ходом.

Так как Петя, получив такие позиции, проиграл, будем считать, что позиции (6, 33) и (8, 32) проигрышные, и игрок, которому они достанутся, проиграет.

Задание 2. Кто из игроков имеет выигрышную стратегию при позициях (6, 32), (7, 32), (8, 31)?

Петя ходит первым, очевидно, что он хочет выиграть, а для этого должен играть правильно. То есть он должен стараться сделать так, чтобы Ване досталась проигрышная позиция. Мы знаем две проигрышные позиции из пункта 1 — (6, 33) и (8, 32). Эти проигрышные позиции Петя может сделать своим первым ходом для Вани из предложенных позиций:

$$(6, 32) \rightarrow (6, 33)$$

$$(7, 32) \rightarrow (8, 32)$$

$$(8, 31) \rightarrow (8, 32)$$



То есть Петя сделал для Вани проигрышную позицию, в результате чего Ваня проиграет, а Петя выиграет своим вторым ходом. Будем считать, что позиции (6, 32), (7, 32), (8, 31) выигрышные, так как они достались Пете и Петя выиграл.

Задание 3. Кто из игроков имеет выигрышную стратегию при позиции (7, 31)?

Рассмотрим все возможные ходы Пети из этой позиции:

(7, 31) $\rightarrow$ (8, 31)

(7, 31) $\rightarrow$ (7, 32)

(7, 31) $\rightarrow$ (14, 31)

(7, 31) $\rightarrow$ (7, 62)

То есть Ване могут достаться позиции (8, 31), (7, 32), (14, 31), (7, 62).

Рассмотрим каждую:

(8, 31) – выигрышная позиция, исходя из пункта 2 решения;

(7, 32) – выигрышная позиция, исходя из пункта 2 решения;

(14, 31) – Ваня выиграет, умножив 31 на 2;

(7, 62) – Ваня выиграет, умножив 62 на 2.

Выходит, что все позиции, которые достались Ване — выигрышные. При этом при позициях (8, 31) и (7, 32) Ване потребуется сделать два хода для победы. То есть при начальной позиции (7, 31) Ваня выиграет максимум двумя ходами.

Задача 2. В электронной таблице записаны матрицы. Необходимо найти произведение двух матриц, а также количество ячеек, больших D8 (результат умножения записан в ячейках, начиная с D6).

	A	B	C	D	E	F	G	H
1	4	5	4	2		1	2	3
2	7	8	9	0		4	2	6
3	6	4	3	4		8	6	2
4	2	0	1	1		4	5	3

Решение:

	A	B	C	D	E	F	G	H
1	4	5	4	2		1	2	3
2	7	8	9	0		4	2	6
3	6	4	3	4		8	6	2
4	2	0	1	1		4	5	3
5								
6	=МУМНОЖ(A1:D4;F1:H4)			64	52	56		
7				111	84	87		
8				62	58	60		
9				14	15	11		



Задача 3. В электронной таблице представлена информация о продуктах. В первой таблице указано количество продуктов на складе на складе (в кг.), во второй – стоимость.

	A	B	C	D	E	F
1	Продукт	Молоко	Сливки	Сыр	Творог	Сметана
2	Количество	55	77	100	150	70
3						
4						
5	Продукт	Сметана	Сыр	Молоко	Сливки	Творог
6	Цена	70	350	60	65	72

Заполните столбец «Цена» первой таблицы данными о стоимости из второй. Запишите формулы для вычислений и получившиеся результаты.

Решение:

8	Продукт	Молоко	Сливки	Сыр	Творог	Сметана	=ГПР(B1;\$B\$5:\$F\$6;2)
9	Количество	55	77	100	150	70	=ГПР(C1;\$B\$5:\$F\$6;2)
10	Цена	60	65	65	72	65	=ГПР(D1;\$B\$5:\$F\$6;2)
11							=ГПР(E1;\$B\$5:\$F\$6;2)
12							=ГПР(F1;\$B\$5:\$F\$6;2)

Задача 4. Символом F обозначен результат логического выражения от 3 аргументов X, Y, Z, а F1 и F2 – обозначение промежуточных результатов:

X	Y	Z	F1	F2	F
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	1	1

Необходимо восстановить выражение по таблице истинности.

Ответ: $(x \wedge y) \rightarrow z \leftrightarrow x$

**Решение:**

1). Первое действие (результат F1) удовлетворяет условиям конъюнкции

2). Второй столбец соответствует результату следования первого действия из z.

3). Результирующий столбец F является результатом равносильности второго действия и x.

В результате получается: $F = (x \wedge y) \rightarrow z \leftrightarrow x$

Задача 5. Выражение $(x \rightarrow y) \vee y \wedge (\overline{z} \rightarrow \overline{y}) \wedge \overline{x}$ ложь. Перечислите выражения, удовлетворяющие такому же набору значений из приведенных ниже:

1). $x \vee \overline{y} \wedge z \rightarrow x$

2). $\overline{y} \wedge x \leftrightarrow z \wedge x$

3). $\overline{x} \rightarrow y \wedge z \vee \overline{x}$

4). $z \leftrightarrow \overline{y} \vee \overline{x} \wedge z$

5). $x \rightarrow y \wedge x \wedge z$

Ответ: 2, 5.

Решение:

В качестве первого шага необходимо построить таблицу истинности:

x	y	z	$x \rightarrow y$	$z \rightarrow y$	$\overline{z} \rightarrow \overline{y}$	$y \wedge \overline{z} \rightarrow y$	$\overline{x}$	$y \wedge \overline{z} \rightarrow y \wedge \overline{x}$	$(x \rightarrow y) \vee y \wedge (\overline{z} \rightarrow \overline{y}) \wedge \overline{x}$
0	0	0	1	1	0	0	1	0	1
0	0	1	1	0	1	0	1	0	1
0	1	0	1	1	0	0	1	0	1
0	1	1	1	1	0	0	1	0	1
1	0	0	0	1	0	0	0	0	0
1	0	1	0	0	1	0	0	0	0
1	1	0	1	1	0	0	0	0	1
1	1	1	1	1	0	0	0	0	1

Ложное значение выражение принимает при значениях:

x	y	z
1	0	0
1	0	1

Необходимо проверить, какие из указанных выше выражений принимают ложное значение при заданных x, y, z. Для этого надо построить 5 таблиц истинности:

x	y	z	$\overline{y}$	$\overline{y} \wedge z$	$x \vee \overline{y} \wedge z$	$x \vee \overline{y} \wedge z \rightarrow x$
1	0	0	1	0	1	1
1	0	1	1	1	1	1

Данный комплект значений равен значению истина для первого выражения, следовательно, первое выражение не подходит. Проверим второе:

x	y	z	$\overline{y}$	$\overline{y} \wedge x$	$\overline{x}$	$z \wedge \overline{x}$	$\overline{y} \wedge x \leftrightarrow z \wedge x$
---	---	---	----------------	-------------------------	----------------	-------------------------	--



1	0	0	1	1	0	0	0
1	0	1	1	1	0	0	0

Выражение №2 при заданных x, y, z принимает значение ложь, значит исходные подходят. Теперь проверяется третье выражение:

x	y	z	$\bar{x}$	$y \wedge z$	$y \wedge z \vee \bar{x}$	$z \wedge \bar{x}$	$\bar{x} \rightarrow y \wedge z \vee \bar{x}$
1	0	0	0	0	0	0	1
1	0	1	0	0	0	0	1

Судя по конечному результату (истина), данный комплект не подходит для выражения 3. Теперь необходимо проверить выражение №4:

x	y	z	$\bar{y}$	$\bar{x}$	$\bar{x} \wedge z$	$\bar{y} \vee \bar{x} \wedge z$	$z \leftrightarrow \bar{y} \vee \bar{x} \wedge z$
1	0	0	1	0	0	1	0
1	0	1	1	0	0	1	1

В результате данного комплекта значений получены значения ложь и истина, выражение не подходит под указанные значения. Осталось проверить последнее выражение:

x	y	z	$y \wedge x$	$\wedge x \wedge z$	$x \rightarrow y \wedge x \wedge z$
1	0	0	0	0	0
1	0	1	0	0	0

Данный комплект значений при подстановке в выражение дал значение ложь, поэтому также является подходящим.

Ответ: 2, 5.

Задача 6. Определите, что будет напечатано в результате работы следующего фрагмента программы:

```
алг
нач
  цел k, s
  s := 10
  k := 2
  нц пока s < 55
    s := s - 2 + k
    k := k + 3
  кц
  вывод k
кон
```

Ответ: 20

**Решение:**

Цикл `while` выполняется до тех пор, пока истинно условие $s < 55$, т.е. переменная s определяет, сколько раз выполнится цикл.

Выпишем все s и k :

s	10	13	19	28	40	55
k	5	8	11	14	17	20

Действие $k=k+2$ выполняется после итерации $s=s-2+k$. Условие $s < 55$ проверяется перед итерациями. Следовательно, ответ 20.

Задача 7. В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет:

<i>Запрос</i>	<i>Количество страниц (тыс.)</i>
<i>Верстальщик</i>	35
<i>WEB-Программист</i>	55
<i>Тестировщик</i>	27
<i>Верстальщик WEB-Программист Тестировщик</i>	90
<i>Тестировщик & WEB-программист</i>	20

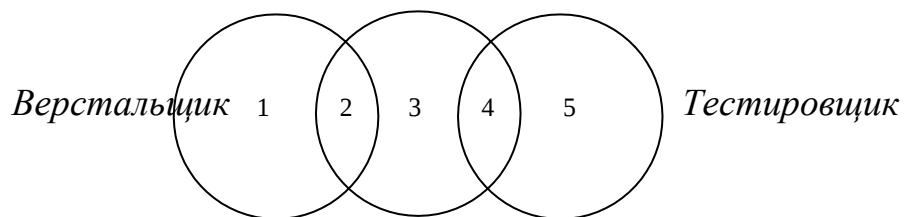
Сколько страниц (в тысячах) будет найдено по запросу **Верстальщик & WEB-программист?**

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

Ответ: 7

**Решение:**

1) построим диаграмму Эйлера-Венна

WEB-программист2) количество сайтов, удовлетворяющих запросу в области i , будем обозначать через N_i

3) 5 областей, известны следующие данные:

$$N_1 + N_2 = 35$$

$$N_4 + N_5 = 27$$

$$N_2 + N_3 + N_4 = 55$$

$$N_1 + N_2 + N_3 + N_4 + N_5 = 90$$

$$N_4 = 20$$

4) необходимо найти область 2:

$$N_5 = 27 - N_4 = 7$$

$$N_1 = 90 - N_5 - (N_2 + N_3 + N_4) = 90 - 7 - 55 = 28$$

$$N_4 = 35 - N_5 = 7$$

Задача 8. Приведен исходный код программы на языках программирования. Во время выполнения 2 итерации было выведено на экран число 2197, необходимо вычислить, какое число было введено в качестве исходного x .



Бейсик	Паскаль
<pre>Dim x As Integer, y As Integer, z As Integer Input x y = 3 z = F(x, 3) End Sub Function F(a As Integer, b As Integer) As Integer Do While a <> 0 If a Mod 2 <> 0 Then n = a ^ b Else n = a \ b End If a = a - 1 Print n Loop End Function</pre>	<pre>Begin var n:integer; While a<>0 do begin If a div 2 <>0 Then n:=exp(b*ln(a)) else n:=a\b; a:=a-1; Writeln(n) End; Begin Var x, y, z: integer; Readln(x); y:=3; z:=F(x,y); End.</pre>
Си	
<pre>#include<stdio.h> int F(int x) { return 19*(x-1)*(x-1); } void main() { int a, b, t, M, R; a = -10; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)>R) { M = t; R = F(t); } } printf("%d", M); }</pre>	

Ответ: 14.

**Решение:**

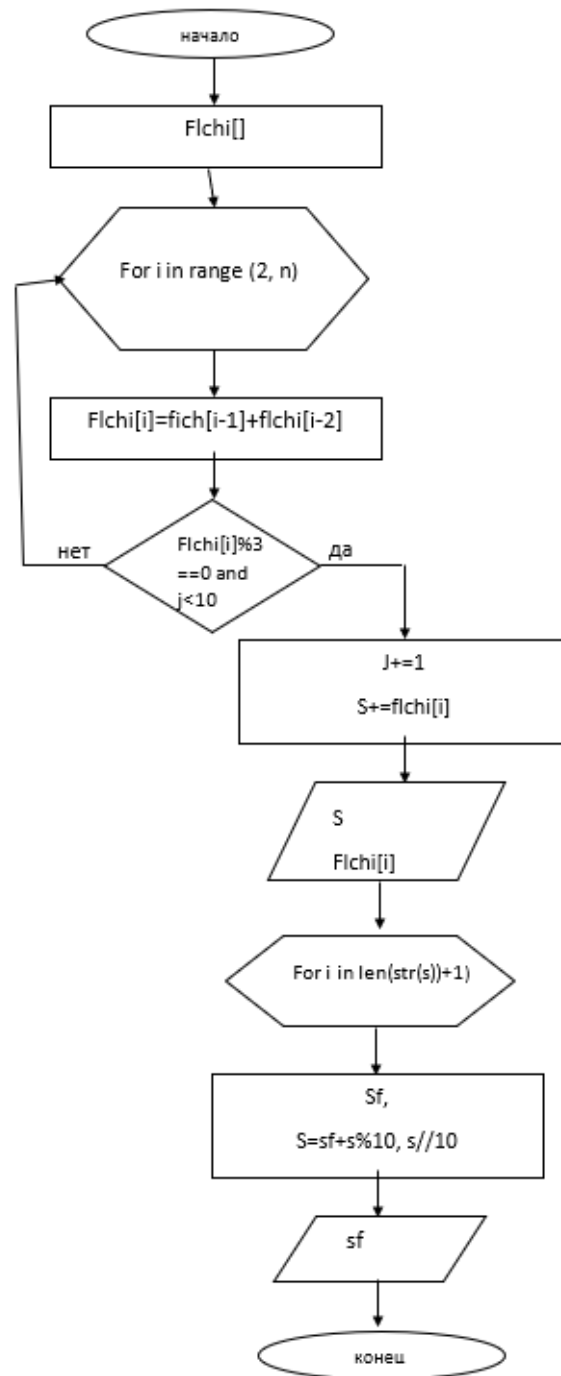
Результат, выведенный на экран (число n) равно 2197 – это третья степень некоторого числа. Для выяснения достаточно взять кубический корень, так как $y=3$, а по условию $n=x^y$. При заданных условиях $x=13$. Так, как это второй шаг выполнения программы необходимо прибавить 1, чтобы определить исходный x (так как шаг цикла $a=a-1$), $x=14$.

Задача 9. Числа Фибоначчи – элементы числовой последовательности: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 610, 987, 1597, 2584, 4181, 6765, ..., в которой каждое последующее число равно сумме двух предыдущих чисел.

Среди чисел Фибоначчи встречаются числа, делящиеся на 3, 5 и 10. Определите сумму первых десяти чисел Фибоначчи, которые делятся на 3 и сумму цифр этих чисел.

Программа должна вывести первые 10 чисел Фибоначчи, делящиеся на 3, сумму этих чисел и сумму цифр этих чисел (не использовать встроенную или библиотечную функции вычисления чисел Фибоначчи).

Решение: в качестве решения сначала приведена блок-схема алгоритма:



Далее приведены примеры решения задачи на языке Python.

Решение с рекурсией (на Python):

```
def fichi(n):                                # Рекурсивная функция вычисления
    числа Фибоначчи
    if n < 2:
        return 1
    else:
        return fichi(n-1)+fichi(n-2)
```



```
n, i, s, sf = 0, 0, 0, 0
print('Первые 10 чисел, делящиеся на 3:')
while True:                                # организуем «вечный» цикл
    n += 1
    f = fichi(n)                            # вызов функции вычисления числа
    Фибоначчи
    if f % 3==0:                            # условие проверки делимости числа на
    3
        i += 1
        print(i, ' число =', f)            # вывод i-го числа
    Фибоначчи согласно условию
        s += f
    if i >=10:                             # условие выхода из «вечного»
    цикла: после вывода первых 10 чисел
        break                             # Фибоначчи, делящихся на 3
print('Сумма первых 10 чисел: s =', s)
for i in range(len(str(s))+1):
    sf, s = sf + s%10, s//10
print('Сумма цифр в этой сумме: sf =', sf)
```

Решение без рекурсии (на Python):

```
j, s, sf, n = 0, 0, 0, 1000                # 1000 – максимальное
    количество чисел Фибоначчи, задаем
fichi = [i for i in range(n)]
fichi[0], fichi[1] = 0, 1
print('Первые 10 чисел Фибоначчи, делящиеся на 3: ')
for i in range(2, n):
    fichi[i] = fichi[i-1] + fichi[i-2]
    if fichi[i]%3 == 0 and j<10:
        j +=1
    print(j, 'число =', fichi[i])
```



```
s += fichi[i]

print('Сумма первых 10 чисел: s =', s)
for i in range(len(str(s))+1):
    sf, s = sf + s%10, s//10
print('Сумма цифр в этой сумме: sf =', sf)
```

```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: D:\Олимпиада_шк_Гранит науки(2020в)\Фиб2020\fib_v3.py ====
Первые 10 чисел, делящиеся на 3:
1 число = 3
2 число = 21
3 число = 144
4 число = 987
5 число = 6765
6 число = 46368
7 число = 317811
8 число = 2178309
9 число = 14930352
10 число = 102334155
Сумма первых 10 чисел: s = 119814915
Сумма цифр в этой сумме: sf = 39
>>>
```

Задача 10. Отдел информационной безопасности выделил диапазон IP-адресов, которые можно использовать для разделения сети предприятия на подсети: 192.168.4.128–192.168.4.159. Дополнительно имеется следующая информация:

- одному устройству должен соответствовать один IP-адрес вида XXX.XXX.XXX.XXX, при этом XXX – число в диапазоне [0:255], называемое октетом;
- при переполнении октета на n единиц, октет получает значение n , а ближайший октет слева от него увеличивает своё значение на 1;
- устройства внутри одной подсети могут беспрепятственно взаимодействовать друг с другом и должны группироваться по их расположению;
- подсети содержат количество адресов, равное степеням двойки: 4, 8, 16, 32, 64;
- сегмент подсети не может начинаться с адреса, не кратного своему размеру. К примеру, подсеть размером в 64 адреса не может начинаться



с адреса 192.168.4.16, а только с адресов: 192.168.4.0, 192.168.4.64, 192.168.4.128...;

- в каждом сегменте сети первый адрес зарезервирован под идентификатор подсети, а последний адрес зарезервирован под широковещательные (broadcast) запросы;
- размер подсети записывается как $32-n$, где n – минимальное количество бит, которое может содержать все номера адресов подсети.

Как оптимальным образом разделить всю сеть на сегменты (подсети) по отделам с указанием идентификатора подсети, размера подсети, broadcast адреса, IP-адреса каждого устройства?

Отделение	Сотрудник	Устройств во
Производственный цех	Степан Белозёров	Автоматизированное рабочее место
		Смартфон
	Никита Устинов	Автоматизированное рабочее место
		Ноутбук
		Смартфон
	Елизавета Владимирова	Автоматизированное рабочее место
		Смартфон
	Фёдор Иванов	Автоматизированное рабочее место
		Смартфон
Планшет		
Администрация	Николай Баранов	Десктоп
		Смартфон
	Людмила Прохорова	Десктоп
		Смартфон
КПП	Павел Зуев	Десктоп
		Смартфон
IT-отдел	Илья Нестеров	Ноутбук
		Смартфон



Решение: Один из возможных вариантов схемы адресации подсетей, при котором подсети охватывают весь диапазон адресов, представлен ниже.

Производственный цех	Степан	Автоматизированное рабочее место	192.168.4.129
192.168.4.128/28	Белозёров	Смартфон	192.168.4.130
Broadcast: 192.168.4.143	Никита Устинов	Автоматизированное рабочее место	192.168.4.131
		Ноутбук	192.168.4.132
		Смартфон	192.168.4.133
	Елизавета Владимировна	Автоматизированное рабочее место	192.168.4.134
		Смартфон	192.168.4.135
	Фёдор Иванов	Автоматизированное рабочее место	192.168.4.136
		Смартфон	192.168.4.137
		Планшет	192.168.4.138
			192.168.4.139
Администрация	Николай	Десктоп	192.168.4.145
192.168.4.144/29	Баранов	Смартфон	192.168.4.146
Broadcast: 192.168.4.151	Людмила	Десктоп	192.168.4.147
	Прохорова	Смартфон	192.168.4.148
			192.168.4.149
			192.168.4.150
КПП	Павел Зуев	Десктоп	192.168.4.153
192.168.4.152/30		Смартфон	192.168.4.154
Broadcast: 192.168.4.155			
IT-отдел	Илья Нестеров	Ноутбук	192.168.4.157
192.168.4.156/30		Смартфон	192.168.4.158
Broadcast: 192.168.4.159			



Задача 11. Как только Пете посчастливилось стать обладателем квартиры, его начал терзать вопрос – что же выгодней, сдавать квартиру или продавать и вкладываться в другую недвижимость на ранних этапах постройки. Петя, будучи программистом, решил подойти к нахождению ответа на данный вопрос профессионально.

Помогите Пете написать программу, которая обладает следующим функционалом:

- На начальном этапе у Пети есть готовая квартира и 0 условных единиц на счету.
- Программа должна хранить информацию о всех квартирах Пети, которые куплены, проданы или сдаются в аренду. Для этого каждой квартире присваивается уникальный идентификатор, а информация по квартире включает в себя:
 - уникальный идентификатор квартиры (например, возрастающий счётчик);
 - стоимость покупки (кроме изначальной квартиры);
 - стоимость продажи (если продана);
 - доход от сдачи в аренду (в месяц);
 - суммарный доход от аренды за всё время сдачи;
 - срок постройки;
 - состояние.
- При вводе команды «list» вся данная информация выводится по каждой квартире.
- Квартира всегда находится в одном из трёх состояний:
 - строится – квартира не может быть продана и не приносит доход от аренды;
 - сдаётся – квартира может быть продана. Если не продана, то приносит за год доход от сдачи в аренду, умноженный на 12;
 - продана – с квартирой ничего нельзя сделать, но информация о ней всё равно должна сохраняться.
- При вводе команды «sell» пользователю выводятся идентификаторы всех квартир в состоянии «Сдаётся». После ввода идентификатора квартиры запрашивается размер денежной суммы за продажу, после чего счёт пользователя пополняется на сумму продажи, а квартира переходит в состояние «Продана».
- При вводе команды «buy» пользователь вводит стоимость новой квартиры, срок её постройки и потенциальный доход от сдачи в аренду на момент, когда она будет построена. Если у пользователя на счету достаточно средств, квартира добавляется в собственность. Если у



пользователя недостаточно средств, выводится сообщение об ошибке.

- При вводе команды «list» выводится список всех квартир, когда-либо бывших в собственности, с информацией.
- При вводе команды «next» происходит пересчёт следующих показателей:
 - счётчик прошедших лет увеличивается на 1;
 - если квартира построена, она приносит годовой доход от сдачи в аренду;
 - если квартира находится на этапе строительства, у неё уменьшается количество оставшихся лет до постройки на 1. Если значение достигло нуля, квартира переходит в статус «Сдаётся» на следующий год.

Затем на экран выводится информация за все прошедшие года:

- баланс пользователя на конец года;
- операции покупки/продажи жилья со стоимостью операции;
- доход от сдачи в аренду, если статус квартиры «сдаётся».

**Решение: пример реализации программы:**

Ввод	Вывод
List	Информация по квартирам: квартира 0: сдаётся в аренду за 20000 в месяц Куплена за None Доход от сдачи в аренду за всё время: 0
Sell	Вы можете продать следующие квартиры: 0 Введите идентификатор квартиры на продажу:
0	Введите стоимость продажи квартиры:
4000000	Квартира 0 была продана за 4000000. Новый баланс: 4000000.
List	Информация по квартирам: квартира 0: продана за 4000000 Куплена за 0 Доход от сдачи в аренду за всё время: 0
Buy	Введите стоимость квартиры:
1500000	Через сколько лет квартира будет построена:
4	Введите стоимость сдачи квартиры в аренду:
20000	Квартира 1 была куплена за 1500000. Новый баланс: 2500000.
Buy	Введите стоимость квартиры:
2000000	Через сколько лет квартира будет построена:
1	Введите стоимость сдачи квартиры в аренду:
15000	Квартира 1 была куплена за 2000000. Новый баланс: 500000.
Next	Квартира 2 построена. Год 1 Квартира 0 была продана за 4000000. Новый баланс: 4000000. Квартира 1 была куплена за 1500000. Новый баланс: 2500000. Квартира 2 была куплена за 2000000. Новый баланс: 500000. Квартира 2 построена. Годовой доход от сдачи квартир: 0 Баланс на конец года: 500000



Next	Год 1 Квартира 0 была продана за 4000000. Новый баланс: 4000000. Квартира 1 была куплена за 1500000. Новый баланс: 2500000. Квартира 2 была куплена за 2000000. Новый баланс: 500000. Квартира 2 построена. Годовой доход от сдачи квартир: 0 Баланс на конец года: 500000 Год 2 Годовой доход от сдачи квартир: 180000 Баланс на конец года: 680000
------	--

Вариант решения на Python:

```
balance = 0
year = 0
STATUS_BUILDING, STATUS_FOR_RENT, STATUS_SOLD = 0, 1, 2
apt_id = 0
apartments = [
    {"status":
        STATUS_FOR_RENT,
        "id": apt_id,
        "sold_for": 0,
        "bought_for": 0,
        "years_to_build": 0,
        "rent_income": 0,
        "rent": 20000}
]
apt_id += 1

full_log
= ""
year_log
= ""

user_input = ""

while user_input != "q":
```



```
user_input = input("Введите команду или 'q' для  
выхода.\n")  
if user_input == "list":  
    print(f'Информация по  
квартирам:') for apt in  
apartments:  
        if apt["status"] ==  
            STATUS_BUILDING: str_status  
            = f'строится, осталось лет:  
{apt["years_to_build"]}'  
        elif apt["status"] == STATUS_FOR_RENT:  
            str_status = f'сдаётся в аренду за  
{apt["rent"]} в месяц'  
        else:  
            str_status = f'продана за {apt["sold_for"]}'  
        print(f''''квартира {apt["id"]}':  
{str_status} Куплена за  
{apt["bought_for"]}  
Доход от сдачи в аренду за всё время:  
{apt["rent_income"]}''')  
  
if user_input ==  
    "sell":  
apartments_to_sell = []  
for apt  
in apartments:  
    if apt["status"] ==  
        STATUS_FOR_RENT:  
apartments_to_sell.append(apt)  
if apartments_to_sell:  
    print("Вы можете продать квартиры со следующими  
идентификаторами:  
")  
for apt in apartments_to_sell:  
    print(f'{apt["id"]} ')  
    print("\n")  
    apt_to_sell_id = int(input("Введите  
идентификатор квартиры на  
продажу: "))  
    apt_price = int(input("Введите стоимость продажи
```



```
квартиры: ") )
for apt in apartments:
    if apt["id"] ==
        apt_to_sell_id:
        apt["status"] =
            STATUS_SOLD
        apt["sold_for"] =
            apt_price
        balance += apt_price
    print(f'Квартира {apt_to_sell_id} была продана
за {apt_price}'.
Новый баланс: {balance}.\n')
        year_log += f'Квартира {apt_to_sell_id} была
        продана за
{apt_price}'. Новый баланс: {balance}.\n'
    else:
        print("Нет готовых квартир для продажи\n")

if user_input == "buy":
    apt_price = int(input("Введите стоимость
квартиры:\n"))
    if balance >= apt_price:
        apt_years_to_build = int(input("Через сколько
лет квартира будет построена:\n"))
        apt_rent_price = int(input("Введите стоимость
сдачи квартиры в аренду:\n"))
        balance -=
            apt_price
        apartments.append(
            {
                "id": apt_id,
                "bought_for":
                    apt_price,
                "years_to_build":
                    apt_years_to_build,
                "status": STATUS_BUILDING,
                "rent":
                    apt_rent_price,
                "rent_income": 0
            })
```



```
print(f'Квартира {apt_id} была куплена за
{apt_price}. Новый баланс: {balance}.\n')
year_log += f'Квартира {apt_id} была куплена за
{apt_price}.\n'
Новый баланс: {balance}.\n'
apt_id
+= 1 else:
    print("Недостаточно средств.\n")

if user_input ==
"next": year +=
1
year_rent_incom
e = 0 for apt
in apartments:
    if apt["status"] ==
STATUS_FOR_RENT:
        apt["rent_income"] +=
        apt["rent"] * 12 balance +=
        apt["rent"] * 12
        year_rent_income +=
        apt["rent"] * 12
    if apt["status"] ==
STATUS_BUILDING:
        apt["years_to_build"]
        -= 1
        if apt["years_to_build"]
        <= 0: apt["status"] =
        STATUS_FOR_RENT
        print(f'Квартира {apt["id"]}
        построена.\n') year_log +=
        f'Квартира {apt["id"]} построена.\n'
    full_log += f''Год {year}
    {year_log}
Годовой доход от сдачи квартир:
{year_rent_income} Баланс на конец
года: {balance}\n\n'''
year_log = ""
print(full_log)
```

**Список рекомендуемой литературы для подготовки**

1. Босова Л.Л., Босова А.Ю., Коломенская Ю.Г. Занимательные задачи по информатике. - М.: БИНОМ. Лаборатория знаний. 2007. – 119 с.
2. Семакин И.Г., Хеннер Е.К., Шеина Т.Ю., Информатика. Базовый уровень: учебник для 11 класса. – 3-е изд. – М.: БИНОМ. Лаборатория знаний, 2014. – 224 с.
3. Брукшир, Дж., Гленн. Введение в компьютерные науки. Общий обзор, 6-е издание. : Пер. с англ. – М. : Издательский дом «Вильямс», 2001. – 688 с.
4. Вовк Е.Т., Информатика: пособие для подготовки к ЕГЭ – М.: БИНОМ. Лаборатория знаний, 2015 – 352 с.
5. Гейн А. Г. Информатика и ИКТ. 10-11 классы. Основы математической логики – М.: «Просвещение». 2012 – 96 с.
6. Гордеев А.В., Молчанов А.Ю. Системное программное обеспечение. СПб.: Питер, 2004. – 416 с.
7. Грэхем Р., Кнут Д., Паташник О. Конкретная математика. Основание информатики: Пер. с англ. – М.: Мир, 1998. –703 с.
8. Долинский М.С. Решение сложных и олимпиадных задач по программированию. – СПб.: Питер, 2006. – 366 с.
9. Ерош И. Л., Сергеев М. Б., Соловьев Н. В. Дискретная математика СПб.: СПбГУАП, 2005. – 144 с.
10. Златопольский Д.М. Занимательная информатика. – М.: Бином. Лаборатория знаний, 2011. – 424 с.
11. Кирюхин В.М., Цветкова М.С. Информатика. Программы внеурочной деятельности учащихся по подготовке к Всероссийской олимпиаде школьников: 5–11 классы. – М.: БИНОМ. Лаборатория знаний, 2014. – 224 с.
12. Кудинов Ю. И., Пащенко Ф. Ф., Келина А. Ю. Практикум по основам современной информатики. – СПб. : Лань, 2011 . – 352 с.
13. Малярчук С. Н. Информатика в определениях, таблицах и схемах. – Харьков: «Ранок». 2011 – 112с.
14. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. 4-е изд. – СПб.: Питер, 2010. – 918 с.
15. Сафронов И.К. Готовимся к ЕГЭ. Информатика. – СПб.: ВHV-Санкт-Петербург. 2009 – 368 с.
16. Соболев Б.В., Галин А.Б. и др., Информатика. – Ростов: «Феникс». 2010 – 446 с.
17. Триумфгородских М. Дискретная математика и математическая логика для информатиков, экономистов и менеджеров. – М.:«Диалог-МИФИ». 2011 – 180 с.

Составители: профессор каф.ИиКТ *Кризский В.Н.*, профессор каф.ИСВТ *Трофимец В.Я.*,

доценты каф. ИиКТ *Е.В. Катунцов, О.В. Косарев, Т.В. Сарпулова*

Научный редактор: зав. каф. ИиКТ, доцент *А.Б. Маховиков*