

**МЕЖРЕГИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ
ИМЕНИ И.Я. ВЕРЧЕНКО**

Профиль:
Информатика и компьютерная безопасность

**Задачи с решениями
(8-10 класс)**

Задача 1. Конференция по компьютерной безопасности	2
Задача 2. Беспилотник	3
Задача 3. DDoS-атака.....	6
Задача 4. Права доступа	6
Задача 5. Blockchain	8

Задача 1. Конференция по компьютерной безопасности

Организация планирует провести конференцию по компьютерной безопасности в городе А. Дата проведения – **16 марта**. Место проведения – большой конференц-зал на **N=2023** посадочных мест.

Для того, чтобы попасть на конференцию требуется входной билет. Организация хочет продать как можно больше билетов. Однако в городе, в котором будет проводиться конференция, продажа билетов регламентирована следующим образом:

1. Объявляется день старта продаж билетов.
2. Каждый день, включая первый день, с 9:00 до 10:00 организатор выставляет на продажу не больше **M = 205** билетов.
3. Каждый день, включая первый день, с 10:00 до 18:00 жители города скупают все выставленные билеты.
4. Каждый день, включая первый день, с 18:00 до 20:00 происходит возврат любых ранее купленных билетов.
5. Продажи заканчиваются за день до начала мероприятия (в день проведения конференции билеты не продаются).

При этом, возможны случаи, когда утром билеты на продажу не выставляются (нет свободных), а вечером происходит возврат ранее купленных билетов.

По опыту прошлых лет, известно, что число возвращаемых билетов с каждым днём растёт и совпадает с последовательностью чисел Фибоначчи.

Помогите организации определить, в какой день следует начать продажи, чтобы было продано максимальное количество билетов, с учетом их возврата. В ответе укажите дату начала продаж и число проданных билетов на момент начала конференции.

Примечание: Числа Фибоначчи – элементы числовой последовательности, в которой первые два числа равны 1 и 1, а каждое последующее число равно сумме двух предыдущих чисел:

$$\begin{aligned} F(1) &= 1, \\ F(2) &= 1, \\ F(n) &= F(n-1) + F(n-2). \end{aligned}$$

Решение

Распишем покупки и возвраты на каждый день с момента старта продаж.

День	Было	Всего куплено	Возврат (число Фибоначчи)	Осталось купленных
1	0	$0 + 205 = 205$	1	204
2	204	$204 + 205 = 409$	1	408
3	408	$408 + 205 = 613$	2	611
4	611	$611 + 205 = 816$	3	813
5	813	$813 + 205 = 1018$	5	1013
6	1013	$1013 + 205 = 1218$	8	1210
7	1210	$1210 + 205 = 1415$	13	1402
8	1402	$1402 + 205 = 1607$	21	1586
9	1586	$1586 + 205 = 1791$	34	1757
10	1757	$1757 + 205 = 1962$	55	1907
11	1907	$1907 + 116 = 2023$	89	1934
12	1934	$1934 + 89 = 2023$	144	1879

Рассмотрим 11й день.

Выставить на продажу нельзя все 205 билетов, поскольку всего осталось: $2023 - 1907 = 116$ (билетов). 11-е число Фибоначчи $34 + 55 = 89$. После возврата останется 1934 билетов (+27 билетов от предыдущего дня).

Теперь рассмотрим 12й день.

Выставить на продажу нельзя все 205 билетов, поскольку всего осталось: $2023 - 1934 = 89$ (билетов). 12-е число Фибоначчи $55 + 89 = 144$.

Следовательно, на 12й день билетов вернут больше, чем можно продать.

Таким образом начинать надо за 11 дней до конференции, а именно 5 марта.

Ответ: 5 марта (за 11 дней до конференции). Всего будет продано 1934 билетов.

Задача 2. Беспилотник

Ивану на Новый Год подарили беспилотник с управлением через специальное приложение на смартфоне, которое ведет журнал отправленных беспилотнику команд.

Для проверки корректности работы беспилотника в инструкции предусмотрен специальный тестовый маршрут, по которому необходимо пролететь с использованием приложения на смартфоне. Иван выполнил все команды управления, пролетел маршрут и вернул беспилотник в исходную точку (см. рисунок).

Маршрут из инструкции:

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G \rightarrow H \rightarrow F \rightarrow E \rightarrow B \rightarrow A$

Координаты точек маршрута из инструкции (X;Y;Z):

A = (0; 0; 0) – исходная точка,

B = (0; 0; 12),

C = (12; 0; 12),

D = (12; 6; 12),

E = (0; 6; 12),

F = (0; 6; 24),

G = (0; 2; 24),

H = (-3; 2; 24).

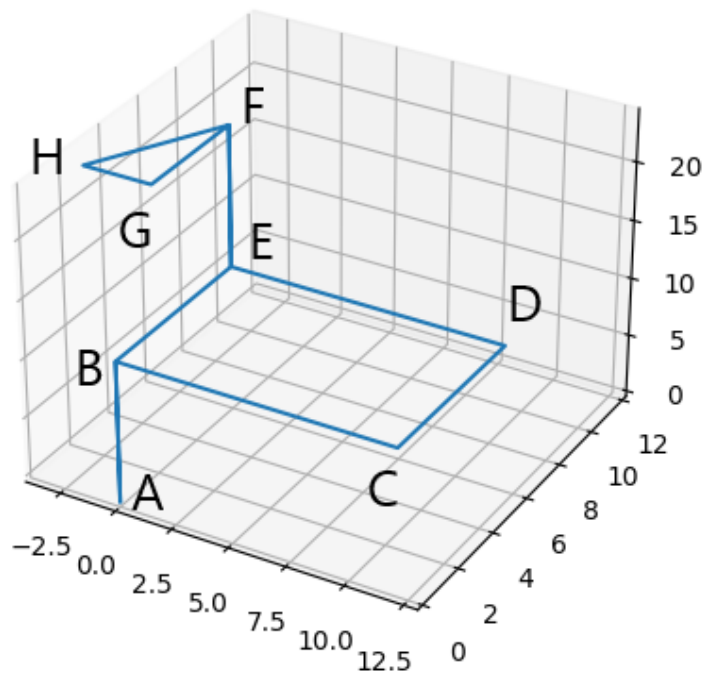


Рисунок – Тестовый маршрут беспилотника (из инструкции).

Единицы изменения шкал – метры

После этого Иван решил самостоятельно управлять беспилотником с использованием приложения. Через какое-то время беспилотник улетел так далеко, что пропал из виду.

На основании журнала отправленных команд помогите Ивану вернуть беспилотник с использованием минимального числа команд. В ответе укажите минимальную последовательность команд, которые необходимо отправить беспилотнику для его возвращения в исходную точку с координатами (0; 0; 0).

Считать, что беспилотник передвигается только по целочисленным координатам, то есть, если после выполнения команды беспилотник должен оказаться в точке с координатами (12,3; 7,8; 5), то он окажется в точке с координатами (12; 8; 5).

К задаче прилагается:

«[drone_test_v1.log](#)» – журнал с командами тестового маршрута из инструкции;

«[drone_v1.log](#)» – журнал с командами, которые отправлял Иван.

Решение

Требуется понять формат команд, а именно:

$id - commandCode - commandParam$

Все части команды записаны в шестнадцатеричном формате.

id – во всех командах одинаковый и равен

вариант 1 – 01AC

вариант 2 – 02C9

Сопоставляя маршрут движения и отправленные команды из журнала, можно выделить следующие команды:

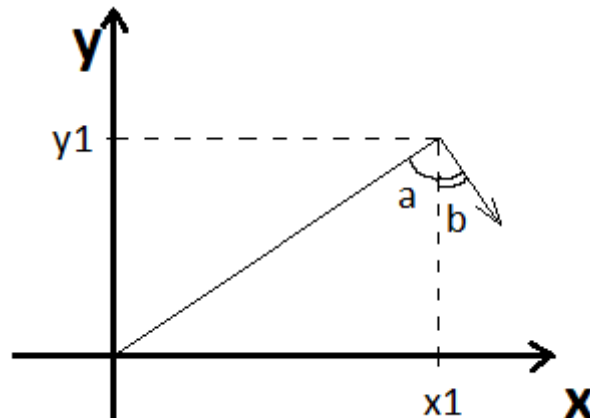
Команда	Вариант 1	Вариант 2
Повернуть налево	18	1C
Повернуть направо	E7	E3
Двигаться вперед	2D	D6
Двигаться вверх	BA	56
Двигаться вниз	45	A9

Проанализируем журнал команд тестового маршрута. Полный тестовый путь беспилотника был следующий:

00: 01AC BA 000C (12₁₀) – *ВВЕРХ 12 м (B)*
 12: 01AC 2D 000C (12₁₀) – *ВПЕРЕД 12 м (C)*
 24: 01AC 18 005A (90₁₀) – *ПОВОРОТ на 90 градусов (C)*
 25: 01AC 2D 0006 (6₁₀) – *ВПЕРЕД 6 м (D)*
 31: 01AC 18 005A (90₁₀) – *ПОВОРОТ на 90 градусов (D)*
 32: 01AC 2D 000C (12₁₀) – *ВПЕРЕД 12 м (E)*
 44: 01AC BA 000C (12₁₀) – *ВВЕРХ 12 м (F)*
 56: 01AC E7 010E (270₁₀) – *ПОВОРОТ* на 270 градусов (F)*
 57: 01AC 2D 0004 (4₁₀) – *ВПЕРЕД 4 м (G)*
 61: 01AC E7 005A (90₁₀) – *ПОВОРОТ* на 90 градусов (G)*
 62: 01AC 2D 0003 (3₁₀) – *ВПЕРЕД 3 м (H)*
 65: 01AC E7 007E (126₁₀) – *ПОВОРОТ* на 126 градусов (H)*
 66: 01AC 2D 0005 (5₁₀) – *ВПЕРЕД 5 м (F)*
 71: 01AC 45 000C (12₁₀) – *ВНИЗ 12 м (E)*
 83: 01AC E7 0090 (220₁₀) – *ПОВОРОТ* на 90 градусов (E)*
 84: 01AC 2D 0006 (6₁₀) – *ВПЕРЕД 6 м (B)*
 90: 01AC 45 000C (12₁₀) – *ВНИЗ 12 м (A)*

Теперь необходимо вычислить координаты, в которых будет находиться беспилотник после каждой команды и угол его поворота, поскольку движение вперед зависит от угла поворота беспилотника относительно своей оси. Допустим, после последней команды беспилотник

оказался в координатах (x_1, y_1, z_1) и «смотрит» под углом β . Тогда ему требуется развернуться на угол $\alpha + \beta$:

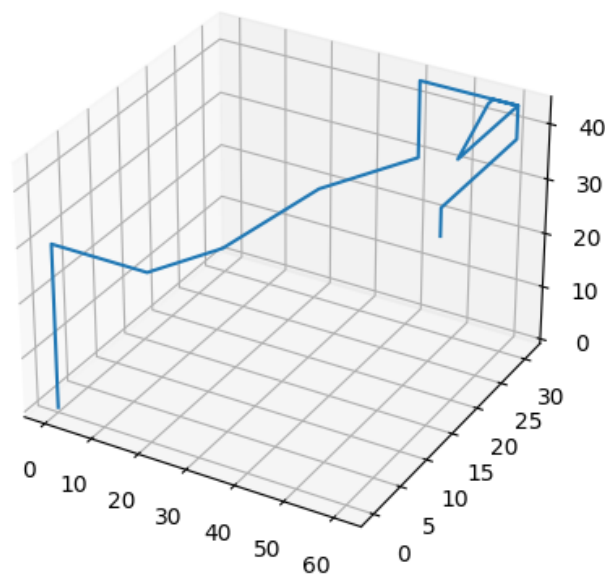


где угол $\alpha = \arctg(y_1/x_1)$.

Угол β изначально равен 90 градусам (поскольку при первом же движении беспилотника прямо, увеличивается только координата x) и далее вычисляется по ходу приема команд.

Итоговый маршрут беспилотника был следующий:

Маршрут дрона



Таким образом ответ будет состоять из команд:

1. Повернуть направо на 49 градусов (E7 31);
2. Двигаться прямо 19 с (2D 13);
3. Спуститься вниз 36 с (45 24).

П.3 может быть как на первом, так и на втором месте.

Ответ:

1. 01ACE731 (направо на 49 градусов);
2. 01AC2D13 (прямо 19 с);
3. 01AC4524 (вниз 36 с).

Задача 3. DDoS-атака

В компании установлен сетевой фильтр, который работает с эффективностью 75% (75% входящих пакетов обрабатываются, остальные 25% накапливаются на сервере и хранятся в течение 24 часов). При накоплении 3 миллионов пакетов сервер перестает работать.

Какое минимальное число пакетов в минуту необходимо отправлять на сервер, чтобы вывести его из строя за время обеденного перерыва, который длится 45 минут, если известно, что в компании работает $N=1200$ сотрудников, каждый из которых в рабочее время получает в среднем $M=245$ пакетов в час? Компания работает по графику 09:00-18:00. Обеденный перерыв с 13:00 до 13:45. Ответ округлите до ближайшего целого.

Решение

Поскольку на сервере накапливается только 25% входящих пакетов, то для накопления 3 миллионов пакетов за сутки, необходимо, чтобы было получено $3\ 000\ 000 / 0.25 = 12\ 000\ 000$ пакетов.

За 4 часа 15 минут предыдущего дня через сетевой фильтр пройдет в среднем $4,25 * 1200 * 245 = 1\ 249\ 500$ пакетов.

Из них 25% или 312 375 пакетов накопятся на сервере.

За 4 рабочих часа текущего дня до обеденного перерыва через сетевой фильтр проходит в среднем

$$4 \text{ часа} * 1200 \text{ сотрудников} * 245 \text{ пакетов} = 1\ 176\ 000 \text{ пакетов.}$$

Из них 25% или 294 000 пакетов накапливаются на сервере.

Таким образом, для вывода сервера из строя необходимо послать на него

$$12\ 000\ 000 - 1\ 249\ 500 - 1\ 176\ 000 = 9\ 574\ 500 \text{ пакетов}$$

за время обеденного перерыва (45 минут).

В одну минуту необходимо передавать как минимум 212 766,67 пакетов.

Ответ: 212 767 пакетов в минуту.

Задача 4. Права доступа

В компании имеется файловое хранилище, на котором записаны $M = 10$ файлов F_1, F_2, \dots, F_{10} . Файловая система хранилища содержит следующие атрибуты доступа к каждому файлу:

R – read (только чтение),

W – write (только запись в файл),

G – grant (предоставление имеющихся прав доступа к файлу другому пользователю),

(K) – число промежуточных каталогов от домашней папки пользователя до места хранения файла.

Предоставление прав происходит следующим образом: если пользователь может только читать файл F (в том числе в результате передачи ему прав), то он может предоставить доступ только на чтение этого файла другому пользователю, но не на запись. При этом для предоставления прав доступа к файлу необходимо к нему обратиться, последовательно просматривая все промежуточные каталоги (просмотр одного каталога занимает 1 единицу времени).

В компании имеется $N = 15$ пользователей U_1, U_2, \dots, U_{15} .

В таблице 1 представлены имеющиеся права доступа пользователей к файлам. В таблице 2 представлена матрица передачи прав доступа к файлам между пользователями.

Таблица 1 – Права доступа пользователей к файлам

	F₁	F₂	F₃	F₄	F₅	F₆	F₇	F₈	F₉	F₁₀
U₁	(6)	(4)	(5)	(3) R	(8)	(10)	(7) W	(4)	(5) W	(3)
U₂	(3) W	(9) R	(6)	(8)	(8)	(2) R	(4)	(3)	(5)	(4) R
U₃	(6)	(6) W	(5)	(7)	(9)	(9)	(2)	(8) R	(6)	(4)
U₄	(5)	(8) R	(7) W	(6)	(8) W	(9)	(6)	(5)	(5)	(3)
U₅	(3)	(3)	(6)	(3) R	(8)	(6)	(5)	(6)	(6)	(2) R
U₆	(6)	(6)	(5)	(8)	(9)	(7) W	(8)	(5)	(5) W	(4)
U₇	(5)	(9)	(6) W	(7)	(10)	(6)	(6)	(4)	(5)	(4)
U₈	(4)	(10)	(7)	(9)	(10) W	(5)	(15) R	(7)	(6)	(3) R
U₉	(7)	(13) W	(5)	(6) R	(9)	(7)	(9)	(6)	(5)	(3)
U₁₀	(5)	(11)	(7)	(5)	(8)	(6)	(10)	(4)	(6)	(4)
U₁₁	(4) R	(7) W	(6)	(6)	(8)	(8) W	(11)	(5)	(6)	(5)
U₁₂	(6)	(4)	(8)	(9) R	(4)	(6)	(8) R	(6)	(7) W	(6)
U₁₃	(9)	(9)	(7) W	(8)	(2) R	(4)	(9)	(8)	(6)	(4)
U₁₄	(5)	(11) R	(6)	(5)	(4)	(9)	(12)	(9)	(7)	(9) R
U₁₅	(6)	(3)	(5)	(7)	(6)	(3)	(8)	(6)	(5) R	(7)

Таблица 2 – Матрица передачи прав доступа к файлам между пользователями (строка – кто имеет право, столбец – кому передает право)

	U₁	U₂	U₃	U₄	U₅	U₆	U₇	U₈	U₉	U₁₀	U₁₁	U₁₂	U₁₃	U₁₄	U₁₅
U₁						G									
U₂										G					
U₃													G		
U₄													G		
U₅												G			
U₆			G					G							
U₇			G												
U₈														G	G
U₉		G													
U₁₀															
U₁₁	G														
U₁₂				G											
U₁₃							G								
U₁₄				G											
U₁₅					G										

Определите, сможет ли пользователь **U₄** получить **права записи** в файл **F₂**? Если да, то за какое число команд предоставления прав доступа и за какое минимальное время произойдет обращение к файлу (интервалами между командами предоставления прав доступа следует пренебречь)? Ответ обоснуйте.

Решение

Все возможные пути до **U₄**

U₉ – **U₂** – **U₁₀** – тупик

(13) – (9) – (11)

U₃ – **U₁₃** – **U₇** – **U₃** – цикл

(6) – (9) – (9) – (6)

U₁₁ – **U₁** – **U₆** – **U₃** – **U₁₃** – **U₇** – **U₃** – цикл

(7) – (4) – (6) – (6) – (9) – (9) – (6)

U₁₁ – **U₁** – **U₆** – **U₈** – **U₁₄** – **U₄** – 46

(7) – (4) – (6) – (10) – (11) – (8) – 46

U₁₁ – **U₁** – **U₆** – **U₈** – **U₁₅** – **U₅** – **U₁₂** – **U₄** – 45

(7) – (4) – (6) – (10) – (3) – (3) – (4) – (8) – 45

Самый короткий путь требует 8 шагов и займет 45 единиц времени.

Ответ: да, за 8 шагов и за 45 единиц времени.

Задача 5. Blockchain

Существует система хранения документов, построенная на основе связного списка блоков (блокчейн). Блоки состоят из набора транзакций и информации о предыдущем блоке цепочки. Каждая транзакция описывает добавляемый в систему один или несколько документов и содержит следующую информацию в формате JSON:

- номер транзакции (`_id`),
- автор документа (`from`),
- адресаты документов (`to`),
- объемы документов (`value`).

Размер транзакции определяется суммарным количеством авторов и адресатов документов, входящих в её состав.

Например, транзакция 11, описывающая документ автора 'user01' для 'user02' объемом 15 страниц

```
{
  _id: 11,
  from: "user01",
  to: [ "user02" ],
  value: [ 15 ]
}
```

имеет размер, равный двум.

Транзакция 12, описывающая два документа автора 'user01': один для 'user02' объемом 15 страниц, второй – для 'user03' объемом 10 страниц

```
{
  _id: 12,
  from: "user01",
  to: [ "user02", "user03" ],
  value: [ 15, 10 ]
}
```

имеет размер, равный трем.

Блок, сформированный на основе транзакций 11 и 12 будет иметь размер, равный $2 + 3 = 5$, и суммарный объем документов, равный $15 + 15 + 10 = 40$.

Все незафиксированные транзакции хранятся в специальном хранилище, откуда их можно взять и добавить в новый блок. В один блок нельзя добавить транзакции с одинаковыми идентификаторами (`_id`). Также у блока есть ограничение на максимальный суммарный размер всех входящих в него транзакций – $S = 15$.

Сформируйте новый блок для добавления в блокчейн из числа незафиксированных транзакций так, чтобы общий объем документов, входящих в его транзакции, был максимальным. В ответе укажите перечень транзакций, входящих в блок, а также общий объем документов из выбранных транзакций.

К задаче прилагается:

«[storage_v1.json](#)» – содержимое хранилища незафиксированных транзакций в формате JSON.

Решение

Для решения задачи необходимо найти такой набор транзакций, сумма размеров которых равна 15, а объем которых максимальный. Для этого можно заранее посчитать размер и объем каждой транзакции, отсортировать по размеру транзакции по убыванию.

ИД транзакции	Размер транзакции	Объем транзакции
4	5	98
11	5	91
26	5	87
25	5	83
19	5	82
17	5	80
15	5	65
29	4	87
18	4	71
28	4	70
20	4	68
10	4	66
2	4	65
24	4	61
1	4	60
7	4	56
8	4	55
27	4	55
30	4	54
23	4	48
21	4	46
5	3	42
6	3	33
12	3	33
13	3	32
16	2	26
22	2	23
3	2	21
9	2	17
14	2	14

Способ 1.

Найдем относительную ценность каждой транзакции, полученную путем деления объема транзакции на ее размер, и отсортируем в порядке убывания ценности.

ID транзакции	Размер транзакции	Объем транзакции	Ценность транзакции
29	4	87	21,75
4	5	98	19,6
11	5	91	18,2
18	4	71	17,75
28	4	70	17,5
26	5	87	17,4
20	4	68	17
25	5	83	16,6
10	4	66	16,5
19	5	82	16,4
2	4	65	16,25
17	5	80	16
24	4	61	15,25
1	4	60	15
7	4	56	14
5	3	42	14
8	4	55	13,75
27	4	55	13,75
30	4	54	13,5
15	5	65	13
16	2	26	13
23	4	48	12
21	4	46	11,5
22	2	23	11,5
6	3	33	11
12	3	33	11
13	3	32	10,67
3	2	21	10,5
9	2	17	8,5
14	2	14	7

Дальше набираем наиболее ценные транзакции так, чтобы суммарный размер был равен 15: ID 29 и 4 дают объем $87+98 = 185$ и размер $4+5 = 9$.

ID 29 может быть заменена комбинацией транзакций с размерами $2 + 2$: ID 3 и ID 9. Тогда суммарный объем будет $21 + 17 = 38$, что меньше объема транзакции с ID 29 (87).

ID 4 может быть заменена комбинацией транзакций с размерами $3 + 2$: ID 6 и ID 3. Тогда суммарный объем будет $33 + 21 = 54$, что меньше объема транзакции с ID 4 (98).

Остается 6 единиц, которые представляют собой комбинации $4+2$ и $3+3$.

Найдем транзакции с таким размером и с максимальной ценностью:

$$4 + 2: \text{ID } 18 \text{ и } 16 - 71 + 26 = 97.$$

$$3 + 3: \text{ID } 5 \text{ и } 6 - 42 + 33 = 75.$$

В итоге получаем следующий набор: (29,4,18,16) – суммарный объем равен 282.

Способ 2

Для поиска максимального объема выбранных транзакций можно написать программу, которая работает по следующему алгоритму.

1. Формируем массив значений объема заполненного блока размерами $1, 2, \dots, 15$: $V = \{V_1, V_2, \dots, V_N\}$. Заполняем его нулями.
2. Берем очередную k -ю транзакцию с размером s_k и объемом v_k , и на её основе формируем новый массив значений объемов блока. Каждый i -й элемент массива равен максимум из следующих значений:
 - предыдущего значения i -й ячейки массива (V_i);
 - объема k -й транзакции v_k , если она помещается в блок размером i ($s_k \leq i$) (то есть вместо того, что было берем текущую k -ю транзакцию, если её размер позволяет её добавить в блок размером i);
 - суммы предыдущего ($i-k$ -го) элемента массива, если $k < i$, плюс объем k -й транзакции v_k (если k предыдущему менее наполненному рюкзаку добавляем k -ю транзакцию и суммарный объем больше текущего).
3. Новый массив значений теперь рассматриваем как текущий массив V .
4. Выполняем пп.2-3, перебирая все имеющиеся транзакции $k=1..30$.
5. Максимальный объем блока будет содержаться в ячейке V_{15} массива значений объема блока.

В итоге получаем следующий набор: (29,4,18,16) – суммарный объем равен 282.

Ответ: Набор транзакций (29,4,18,16) – суммарный объем равен 282.
