



**МАТЕРИАЛЫ ЗАДАНИЙ МЕЖРЕГИОНАЛЬНОЙ ОЛИМПИАДЫ  
ШКОЛЬНИКОВ ИМЕНИ И.Я. ВЕРЧЕНКО ПО ПРОФИЛЮ  
«КОМПЬЮТЕРНАЯ БЕЗОПАСНОСТЬ»  
(2023/2024 УЧЕБНЫЙ ГОД)**

**ЗАКЛЮЧИТЕЛЬНЫЙ ЭТАП  
8-10 КЛАССЫ**

**СОДЕРЖАНИЕ**

Задача 1. Компьютерная игра.....	2
Задача 2. Планировщик.....	6
Задача 3. Хеш-пароль.....	11
Задача 4. Номер телефона.....	14
Задача 5. Скидка.....	16

## Задача 1. Компьютерная игра

### Вариант 1

В компьютерной игре общий рейтинг команды определяется набором персонажей и их влиянием друг на друга. Каждый персонаж может влиять на рейтинг команды как положительно, так и отрицательно. Для вычисления общекомандного рейтинга необходимо учесть взаимные веса всех возможных пар персонажей внутри команды.

Например, рассмотрим таблицу взаимных весов персонажей.

	Персонаж 1	Персонаж 2	Персонаж 3	Персонаж 4
Персонаж 1		-3	+5	+2
Персонаж 2	-3		+1	-4
Персонаж 3	+5	+1		-6
Персонаж 4	+2	-4	-6	

Если команда состоит из двух персонажей, то наибольший рейтинг будет равен 5 у команды, состоящей из Персонажа 1 и Персонажа 3.

Если в команду будут входить Персонаж 1 (П1), Персонаж 2 (П2) и Персонаж 3 (П3), то общекомандный рейтинг будет равен сумме рейтингов всех пар персонажей команды: (П1-П2), (П1-П3), (П2-П3):

$$-3 + 5 + 1 = 3.$$

На основе таблицы взаимных весов персонажей определите команду из трех персонажей, которая будет иметь **НАИБОЛЬШИЙ** общекомандный рейтинг. В ответе укажите имена персонажей и значение общекомандного рейтинга.

	Странник	Дракон	Эльф	Титан	Воин	Король	Ведьма	Старец	Волшебница	Рыцарь
Странник		-17	+2	-10	+13	-4	-5	-4	+6	-1
Дракон	-17		+11	+4	+3	+15	+7	-2	-5	+8
Эльф	+2	+11		-4	-19	+5	-10	+1	+14	-2
Титан	-10	+4	-4		+4	+7	+15	-3	-5	+12
Воин	+13	+3	-19	+4		+9	+10	+6	-9	+4
Король	-4	+15	+5	+7	+9		-5	-2	+3	-7
Ведьма	-5	+7	-10	+15	+10	-5		+3	+10	+4
Старец	-4	-2	+1	-3	+6	-2	+3		-1	-8
Волшебница	+6	-5	+14	-5	-9	+3	+10	-1		+6
Рыцарь	-1	+8	-2	+12	+4	-7	+4	-8	+6	

### Решение

Необходимо перебрать все возможные комбинации из трех персонажей (их всего 120). Для каждой посчитать общий рейтинг и найти команду с максимальным рейтингом.

Для упрощения поиска можно написать программу, перебирающую все возможные комбинации и выводящую максимальное значение рейтинга на экран.

Листинг на языке C++	Листинг на языке Python
<pre>#include &lt;iostream&gt; using namespace std;  int main() {     int A[10][10]={         {0, -17, 2, -10, 13, -4, -5, -4, 6, -1},         {-17, 0, 11, 4, 3, 15, 7, -2, -5, 8},</pre>	<pre>A = [     [0, -17, 2, -10, 13, -4, -5, -4, 6, -1],     [-17, 0, 11, 4, 3, 15, 7, -2, -5, 8],     [2, 11, 0, -4, -19, 5, -10, 1, 14, -2],     [-10, 4, -4, 0, 4, 7, 15, -3, -5, 12],     [13, 3, -19, 4, 0, 9, 10, 6, -9, 4],     [-4, 15, 5, 7, 9, 0, -5, -2, 3, -7],     [-5, 7, -10, 15, 10, -5, 0, 3, 10, 4],     [-4, -2, 1, -3, 6, -2, 3, 0, -1, -8],</pre>

<pre> {2,11,0,-4,-19,5,-10,1,14,-2}, {-10,4,-4,0,4,7,15,-3,-5,12}, {13,3,-19,4,0,9,10,6,-9,4}, {-4,15,5,7,9,0,-5,-2,3,-7}, {-5,7,-10,15,10,-5,0,3,10,4}, {-4,-2,1,-3,6,-2,3,0,-1,-8}, {6,-5,14,-5,-9,3,10,-1,0,6}, {-1,8,-2,12,4,-7,4,-8,6,0} }; int sumMax=-100, sum, p1,p2,p3; for (int i=0; i&lt;10; i++)     for (int j=0; j&lt;10; j++)         for (int k=0;k&lt;10;k++)             if ((i!=j) &amp;&amp; (i!=k) &amp;&amp; (j!=k))                 {                     sum=A[i][j]+A[i][k]+A[j][k];                     if (sum &gt; sumMax) {                         sumMax = sum;                         p1 = i + 1;                         p2 = j + 1;                         p3 = k + 1;                     }                 } } cout &lt;&lt; "Сумма " &lt;&lt; sumMax &lt;&lt; "\nПерсонажи " &lt;&lt; p1 &lt;&lt; ", " &lt;&lt; p2 &lt;&lt; ", " &lt;&lt; p3; } </pre>	<pre> [6,-5,14,-5,-9,3,10,-1,0,6], [-1,8,-2,12,4,-7,4,-8,6,0]] sumMax = -100 p1, p2, p3 = 0, 0, 0 for i in range(10):     for j in range(10):         for k in range(10):             if i!=j and i!=k and j!=k:                 sum=A[i][j]+A[i][k]+A[j][k]                 if sum &gt; sumMax:                     sumMax = sum                     p1 = i + 1                     p2 = j + 1                     p3 = k + 1 print(f"Сумма {sumMax}") print(f"Персонажи {p1}, {p2}, {p3}") </pre>
---	---

Результат выполнения программы:

Сумма 31  
Персонажи 2, 3, 6

**Ответ:**     **1. Максимальная сумма – 31**  
              **2. Персонажи: Дракон, Эльф и Король**

**Вариант 2**

В компьютерной игре общий рейтинг команды определяется набором персонажей и их влиянием друг на друга. Каждый персонаж может влиять на рейтинг команды как положительно, так и отрицательно. Для вычисления общекомандного рейтинга необходимо учесть взаимные веса всех возможных пар персонажей внутри команды.

Например, рассмотрим таблицу взаимных весов персонажей.

	Персонаж 1	Персонаж 2	Персонаж 3	Персонаж 4
Персонаж 1		-3	+5	+2
Персонаж 2	-3		+1	-4
Персонаж 3	+5	+1		-6
Персонаж 4	+2	-4	-6	

Если команда состоит из двух персонажей, то наибольший рейтинг будет равен 5 у команды, состоящей из Персонажа 1 и Персонажа 3.

Если в команду будут входить Персонаж 1 (П1), Персонаж 2 (П2) и Персонаж 3 (П3), то общекомандный рейтинг будет равен сумме рейтингов всех пар персонажей команды: (П1-П2), (П1-П3), (П2-П3):

$$-3 + 5 + 1 = 3.$$

На основе таблицы взаимных весов персонажей определите команду из трех персонажей, которая будет иметь **НАИМЕНЬШИЙ** общекомандный рейтинг. В ответе укажите имена персонажей и значение общекомандного рейтинга.

	Странник	Дракон	Эльф	Титан	Воин	Король	Ведьма	Старец	Волшебница	Рыцарь
Странник		+15	+13	-11	-4	+2	-12	-17	-12	+9
Дракон	+15		-4	-6	-10	+15	-13	-9	-14	-5
Эльф	+13	-4		-20	-1	-1	-11	-18	+2	-18
Титан	-11	-6	-20		-3	-8	+7	+4	+6	+1
Воин	-4	-10	-1	-3		+7	+6	-10	-2	-9
Король	+2	+15	-1	-8	+7		+19	+11	+8	+5
Ведьма	-12	-13	-11	+7	+6	+19		+18	-12	-13
Старец	-17	-9	-18	+4	-10	+11	+18		+16	+14
Волшебница	-12	-14	+2	+6	-2	+8	-12	+16		-16
Рыцарь	+9	-5	-18	+1	-9	+5	-13	+14	-16	

### Решение

Необходимо перебрать все возможные комбинации из трех персонажей (их всего 120). Для каждой посчитать общий рейтинг и найти команду с максимальным рейтингом.

Для упрощения поиска можно написать программу, перебирающую все возможные комбинации и выводящую максимальное значение рейтинга на экран.

Листинг на языке C++	Листинг на языке Python
<pre>#include &lt;iostream&gt; using namespace std;  int main() {     int A[10][10]={         {0,15,13,-11,-4,2,-12,-17,-12,9},         {15,0,-4,-6,-10,15,-13,-9,-14,-5},         {13,-4,0,-20,-1,-1,-11,-18,2,-18},         {-11,-6,-20,0,-3,-8,7,4,6,1},         {-4,-10,-1,-3,0,7,6,-10,-2,-9},         {2,15,-1,-8,7,0,19,11,8,5},         {-12,-13,-11,7,6,19,0,18,-12,-13},         {-17,-9,-18,4,-10,11,18,0,16,14},         {-12,-14,2,6,-2,8,-12,16,0,-16},         {9,-5,-18,1,-9,5,-13,14,-16,0}     };     int sumMin=100, sum, p1,p2,p3;     for (int i=0; i&lt;10; i++)         for (int j=0; j&lt;10; j++)             for (int k=0;k&lt;10;k++)                 if ((i!=j) &amp;&amp; (i!=k) &amp;&amp; (j!=k))                 {                     sum=A[i][j]+A[i][k]+A[j][k];                     if (sum &lt; sumMin) {                         sumMin = sum;                         p1 = i + 1;                         p2 = j + 1;                         p3 = k + 1;                     }                 }     cout &lt;&lt; "Сумма " &lt;&lt; sumMin &lt;&lt;     "\nПерсонажи " &lt;&lt; p1 &lt;&lt; ", " &lt;&lt; p2 &lt;&lt; ", "     &lt;&lt; p3; }</pre>	<pre>A = [     [0,15,13,-11,-4,2,-12,-17,-12,9],     [15,0,-4,-6,-10,15,-13,-9,-14,-5],     [13,-4,0,-20,-1,-1,-11,-18,2,-18],     [-11,-6,-20,0,-3,-8,7,4,6,1],     [-4,-10,-1,-3,0,7,6,-10,-2,-9],     [2,15,-1,-8,7,0,19,11,8,5],     [-12,-13,-11,7,6,19,0,18,-12,-13],     [-17,-9,-18,4,-10,11,18,0,16,14],     [-12,-14,2,6,-2,8,-12,16,0,-16],     [9,-5,-18,1,-9,5,-13,14,-16,0] ] sumMin = 100 p1, p2, p3 = 0, 0, 0 for i in range(10):     for j in range(10):         for k in range(10):             if i!=j and i!=k and j!=k:                 sum=A[i][j]+A[i][k]+A[j][k]                 if sum &lt; sumMin:                     sumMin = sum                     p1 = i + 1                     p2 = j + 1                     p3 = k + 1 print(f"Сумма {sumMin}") print(f"Персонажи {p1}, {p2}, {p3}")</pre>

Результат выполнения программы:

Сумма -42

Персонажи 3, 7, 10

**Ответ:**

- 1. Минимальная сумма – -42**
- 2. Персонажи: Эльф, Ведьма и Рыцарь**

---

## Задача 2. Планировщик

### Вариант 1

В операционной системе планировщик формирует очередь выполнения задач по ближайшему времени завершения. В первую очередь запускается задача, на выполнение которой требуется меньше всего времени, а самые длительные задачи выполняются в последнюю очередь. Задачи из очереди могут запускаться одновременно при условии, что общая нагрузка ЦП не будет превышать 90% ( $\leq 90\%$ ). Если следующая из очереди задача не может быть запущена из-за загрузки ЦП, она не может быть пропущена. Она сможет запуститься только после завершения предыдущих задач и снижения нагрузки на ЦП.

В таблице представлен список всех запланированных задач со сроком выполнения (в формате минуты:секунды) и нагрузкой на ЦП (в процентах).

№ задачи	Время выполнения	Загрузка ЦП
1	03:20	40
2	02:15	50
3	05:24	70
4	01:15	20
5	09:22	60
6	02:30	55
7	03:15	40
8	02:20	35
9	01:49	65
10	02:00	50
11	03:13	45
12	02:45	75
13	07:15	25
14	01:55	60
15	01:01	55
16	06:45	80
17	01:40	45
18	04:02	65
19	02:18	15
20	03:00	50

Определите, сколько времени (в секундах) потребуется для выполнения всех задач. Время запуска задач и другие накладные расходы не учитываются.

### Решение

На первом шаге необходимо отсортировать все процессы по времени выполнения от наименьшего до наибольшего.

№ задачи	Время выполнения	В секундах	Загрузка ЦП
15	1:01	61	55
4	1:15	75	20
17	1:40	100	45
9	1:49	109	65
14	1:55	115	60
10	2:00	120	50

2	2:15	135	50
19	2:18	138	15
8	2:20	140	35
6	2:30	150	55
12	2:45	165	75
20	3:00	180	50
11	3:13	193	45
7	3:15	195	40
1	3:20	200	40
18	4:02	242	65
3	5:24	324	70
16	6:45	405	80
13	7:15	435	25
5	9:22	562	60

Далее необходимо распределить все задачи по времени с учетом возможности параллельного выполнения.

Отсчет времени	Событие	Длительность события (секунды)	Требуемая загрузка ЦП	Общая загрузка ЦП
0 с	Запуск задачи 15	61	55	55
0 с	Запуск задачи 4	75	20	$55+20 = 75$
61 с	Завершение задачи 15			20
61 с	Запуск задачи 17	100	45	$20+45=65$
75 с	Завершение задачи 4			45
161 с	Завершение задачи 17			0
161 с	Запуск задачи 9	109	65	65
270 с	Завершение задачи 9			0
270 с	Запуск задачи 14	115	60	60
385 с	Завершение задачи 14			0
385 с	Запуск задачи 10	150	50	50
505 с	Завершение задачи 10			0
505 с	Запуск задачи 2	135	50	50
505 с	Запуск задачи 19	138	15	$50+15 = 65$
640 с	Завершение задачи 2			15
640 с	Запуск задачи 8	140	35	$15+35 = 50$
643 с	Завершение задачи 19			35
643 с	Запуск задачи 6	150	55	$35+55 = 90$
780 с	Завершение задачи 8			55
793 с	Завершение задачи 6			0
793 с	Запуск задачи 12	165	75	75
958 с	Завершение задачи 12			0
958 с	Запуск задачи 20	180	50	50
1138 с	Завершение задачи 20			0
1138 с	Запуск задачи 11	193	45	45
1138 с	Запуск задачи 7	195	40	$45+40=85$

1331 с	Завершение задачи 11			40
1331 с	Запуск задачи 1	200	40	40+40=80
1333 с	Завершение задачи 7			40
1531 с	Завершение задачи 1			0
1531 с	Запуск задачи 18	242	65	65
1773 с	Завершение задачи 18			0
1773	Запуск задачи 3	324	70	70
2097 с	Завершение задачи 3			0
2097 с	Запуск задачи 16	405	80	80
2502 с	Завершение задачи 16			0
2502 с	Запуск задачи 13	435	25	25
2502 с	Запуск задачи 5	562	60	25+60=85
2937 с	Завершение задачи 13			60
3064 с	Завершение задачи 5			0

Обработка всех процессов займет 3064 секунды.

**Ответ: 3064 секунды.**

### **Вариант 2**

В операционной системе планировщик формирует очередь выполнения задач по ближайшему времени завершения. В первую очередь запускается задача, на выполнение которой требуется меньше всего времени, а самые длительные задачи выполняются в последнюю очередь. Задачи из очереди могут запускаться одновременно при условии, что общая нагрузка ЦП не будет превышать 90% ( $\leq 90\%$ ). Если следующая из очереди задача не может быть запущена из-за загрузки ЦП, она не может быть пропущена. Она сможет запуститься только после завершения предыдущих задач и снижения нагрузки на ЦП.

В таблице представлен список всех запланированных задач со сроком выполнения (в формате минуты:секунды) и нагрузкой на ЦП (в процентах).

№ задачи	Время выполнения	Загрузка ЦП
1	04:02	50
2	01:17	45
3	02:55	55
4	03:00	60
5	04:10	40
6	01:00	65
7	02:24	60
8	01:39	70
9	03:24	35
10	01:22	40
11	02:45	35
12	04:06	80
13	01:18	75
14	02:29	30

15	03:30	50
16	01:10	45
17	06:14	85
18	02:11	65
19	01:50	35
20	02:36	70

Определите, сколько времени (в секундах) потребуется для выполнения всех задач. Время запуска задач и другие накладные расходы не учитываются.

### Решение

На первом шаге необходимо отсортировать все процессы по времени выполнения от наименьшего до наибольшего.

№ задачи	Время выполнения	В секундах	Загрузка ЦП
6	1:00	60	65
16	1:10	70	45
2	1:17	77	45
13	1:18	78	75
10	1:22	82	40
8	1:39	99	70
19	1:50	110	35
18	2:11	131	65
7	2:24	144	60
14	2:29	149	30
20	2:36	156	70
11	2:45	165	35
3	2:55	175	55
4	3:00	180	60
9	3:24	204	35
15	3:30	210	50
1	4:02	242	50
12	4:06	246	80
5	4:10	250	40
17	6:14	374	85

Далее необходимо распределить все задачи по времени с учетом возможности параллельного выполнения.

Далее необходимо распределить все процессы по времени

Время	Процессы	Загрузка ЦП
60 секунд	6	65
70 секунд	16,2	90
7 секунд	2	45
78 секунд	13	75
82 секунды	10	40
99 секунд	8	70
110 секунд	19	35
131 секунда	18	65
144 секунды	7,14	90

5 секунд	14	30
156 секунд	20	70
165 секунд	11,3	90
10 секунд	3	55
180 секунд	4	60
204 секунды	9,15	85
6 секунд	15	50
242 секунды	1	50
246 секунд	12	80
250 секунд	5	40
374 секунды	17	85

Обработка всех процессов займет 2619 секунд.

**Ответ: 2619 секунд.**

---

### Задача 3. Хеш-пароль

#### Вариант 1

Администратор Петр забыл пароль от своей учетной записи. Но у него имеется документ с предыдущими паролями. Из системы ему удалось получить сохраненные хеш-строки предыдущих паролей, которые Петр сопоставил с имеющимися у него паролями (см. таблицу).

Пароль	Хеш-строка
welc0mehome222	ap8f9dpwudp777
h1st0ryb00k	w8yc9xzo99t
n4tur3phot0graphy	n5ckx6bwuc9rxhbwz
cod3c0mplete	fus6f9dbepcp
???	n6aycxunrb5yua9xs

Политикой безопасности системы предусмотрено, что пароль может содержать только строчные латинские буквы и цифры 0-9.

На основании представленных паролей и их хеш-строк определите алгоритм хеширования и восстановите пароль.

#### Решение

Если внимательно посмотреть на пароли и их хеш-строки, то можно отметить, что:

- длина пароля и длина хеш-строки совпадают;
- буквы пароля заменяются на буквы в хеш-строке, а цифры – на цифры соответственно;
- одна и та же буква/цифра в пароле всегда соответствует одной и той же букве/цифре в хеш-строке.

Отсюда можно сделать вывод о том, что хеш-строка была получена методом подстановки, где каждый символ пароля заменяется на некоторый фиксированный символ. Используя имеющиеся пароли и хеш-строки, можно восстановить часть алфавита.

Оригинальный алфавит	Алфавит для шифрования
w	a
e	p
c	f
m	d
h	w
o	u
s	y
t	c
r	x
y	z
k	t
n	n
u	k
p	b
g	r
a	h
d	s
l	e
0	9
1	8
2	7
3	6
4	5

С помощью полученной части алфавита можно восстановить исходный пароль, все необходимые символы у нас есть.

n6aycxunrb5yya9xs – n3wstrongp4ssw0rd

**Ответ: n3wstrongp4ssw0rd**

### Вариант 2

Администратор Петр забыл пароль от своей учетной записи. Но у него имеется документ с предыдущими паролями. Из системы ему удалось получить сохраненные хеш-строки предыдущих паролей, которые Петр сопоставил с имеющимися у него паролями (см. таблицу).

Пароль	Хеш-строка
wllllamsh4k3sp3are	e1pplykwz4t3wv3yrx
crypt0n1t3	frdva0i1a3
b10ckh4int4b13	lp0ftz4qia4lp3
am0rt1zedqu3u3	yk0ra1hxobg3g3
???	be3rad1zxpp0e0rpo

Политикой безопасности системы предусмотрено, что пароль может содержать только строчные латинские букв и цифры 0-9.

На основании представленных паролей и их хеш-строк определите алгоритм хеширования и восстановите пароль.

### Решение

Если внимательно посмотреть на пароли и их хеш-строки, то можно отметить, что:

- длина пароля и длина хеш-строки совпадают;
- буквы пароля заменяются на буквы в хеш-строке, а цифры – на цифры соответственно;
- одна и та же буква/цифра в пароле всегда соответствует одной и той же букве/цифре в хеш-строке.

Отсюда можно сделать вывод о том, что хеш-строка была получена методом подстановки, где каждый символ пароля заменяется на некоторый фиксированный символ. Используя имеющиеся пароли и хеш-строки, можно восстановить часть алфавита.

Оригинальный алфавит	Алфавит для шифрования
w	e
l	p
a	y
m	k
s	w
h	z
k	t
p	v
r	r
e	x
c	f
y	d
t	a
b	l
i	q
n	i
z	h

d	o
q	b
u	g
0	0
1	1
3	3
4	4

С помощью полученной части алфавита можно восстановить исходный пароль, все необходимые символы у нас есть.

be3rad1zxpp0e0rpo - qw3rty1hell0w0rld

**Ответ: qw3rty1hell0w0rld**

---

## Задача 4. Номер телефона

### Вариант 1

На конференции Николай познакомился с руководителем компании, в которой всегда мечтал работать, и взял у него визитную карточку. Но через некоторое время обнаружил, что потерял её. Николай запомнил код оператора и некоторые особенности остальной части номера телефона, состоящей из 7 цифр:

– первые три цифры номера представляют собой арифметическую прогрессию с шагом, не равным 0 (арифметической прогрессией называют числовую последовательность, каждый последующий член которой равен предшествующему, сложенному с одним и тем же числом – шагом прогрессии);

– последние три цифры являются перестановкой первых трёх цифр, но не равны им (то есть те же самые цифры, но в другом порядке);

– цифра посередине номера не равна ни одной из цифр в первой и последней тройке.

Сколько номеров придется перебрать Николаю, чтобы гарантированно узнать номер телефона, который был указан на визитке?

### Решение

#### Способ 1. Аналитический

Известно, что номер телефона состоит из цифр от 0 до 9.

Найдем все возможные комбинации для первых трех чисел. В условии не указано является ли арифметическая прогрессия возрастающей или убывающей, поэтому будет рассматривать оба случая.

Шаг 1: 012, 123, 234, 345, 456, 567, 678, 789

Шаг 2: 024, 135, 246, 357, 468, 579

Шаг 3: 036, 147, 258, 369

Шаг 4: 048, 159

Шаг -1: 987, 876, 765, 654, 543, 432, 321, 210

Шаг -2: 975, 864, 753, 642, 531, 420

Шаг -3: 963, 852, 741, 630

Шаг -4: 951, 840

Всего комбинаций – 40.

Последние три цифры являются перестановкой первых трех цифр. Всего возможно 6 перестановок для любой комбинации из 3 неповторяющихся цифр. В нашем случае их может быть 5, потому что по условию последние цифры повторяют только значения первых трех, но не их порядок.

Средняя цифра может принимать одно из 7 значений, не равное цифрам из первой и второй тройки.

Таким образом, общее число номеров  $40 \cdot 5 \cdot 7 = 1400$ .

#### Способ 2. Программный

Напишем программу, которая формирует арифметическую прогрессию из трех цифр так, чтобы все значения прогрессии были меньше 10 и больше либо равны 0. Учтем, что прогрессия может быть как возрастающей, так и убывающей.

Всего комбинаций – 40.

Последние три цифры являются перестановкой первых трех цифр. Всего возможно 6 перестановок для любой комбинации из 3 неповторяющихся цифр. В нашем случае их может быть 5, потому что по условию последние цифры повторяют только значения первых трех, но не их порядок.

Средняя цифра может принимать одно из 7 значений, не равное цифрам из первой и второй тройки. Таким образом, общее число номеров  $40 \cdot 5 \cdot 7 = 1400$ .

**Ответ:  $1400 - 1 = 1399$ .**

## Вариант 2

На конференции Николай познакомился с руководителем компании, в которой всегда мечтал работать, и взял у него визитную карточку. Но через некоторое время обнаружил, что потерял её. Николай запомнил код оператора и некоторые особенности остальной части номера телефона, состоящей из 7 цифр:

- первые три цифры являются последовательностью неповторяющихся четных чисел;
- последние три цифры являются перестановкой первых трёх цифр, но не равны им (то есть те же самые цифры, но в другом порядке);
- цифра посередине номера не равна ни одной из цифр в первой и последней тройке.

Сколько номеров придется перебрать Николаю, чтобы гарантированно узнать номер телефона, который был указан на визитке?

## Решение

### Способ 1. Аналитический

Известно, что номер телефона состоит из цифр от 0 до 9.

Найдем все возможные комбинации для первых трех цифр. Всего нечетных цифр 5 (0,2,4,6,8). Количество комбинаций из неповторяющихся трех цифр равно  $5 \cdot 4 \cdot 3 = 60$ .

Последние три цифры являются перестановкой первых трех цифр. Всего возможно 6 перестановок для любой комбинации из 3 неповторяющихся цифр. В нашем случае их может быть 5, потому что по условию последние цифры повторяют только значения первых трех, но не их порядок.

Средняя цифра может принимать одно из 7 значений, не равное цифрам из первой и второй тройки.

Таким образом, общее число номеров  $60 \cdot 5 \cdot 7 = 2100$ .

### Способ 2. Программный

Напишем программу, которая формирует последовательность из трех четных неповторяющихся цифр.

Результат работы программы:

024, 026, 028, 042, 046, 048, 062, 064, 068, 082, 084, 086, 204, 206, 208,  
240, 246, 248, 260, 264, 268, 280, 284, 286, 402, 406, 408, 420, 426, 428,  
460, 462, 468, 480, 482, 486, 602, 604, 608, 620, 624, 628, 640, 642, 648,  
680, 682, 684, 802, 804, 806, 820, 824, 826, 840, 842, 846, 860, 862, 864

Всего комбинаций – 60

Последние три цифры являются перестановкой первых трех цифр. Всего возможно 6 перестановок для комбинации из 3 чисел. В нашем случае их может быть 5, потому что по условию последние цифры повторяют только значения первых трех, но не их порядок.

Средняя цифра может принимать одно из 7 значений, не равное цифрам из первой и второй тройки. Таким образом, общее число номеров  $60 \cdot 5 \cdot 7 = 2100$ .

**Ответ: 2100-1 = 2099.**

## Задача 5. Скидка

### Вариант 1

Магазин придумал свою систему скидок на весь 2024 год. Скидка (в процентах) вычисляется на основе даты по следующей формуле:

$$C = \frac{(D\%10) * (M\%3) * C_D}{M + (\text{mod}(30 - N) + 1)}$$

где

$D$  – день месяца (от 1 до 28..31 в зависимости от месяца);

$M$  – порядковый номер месяца (от 1 до 12);

$N$  – порядковый номер текущей недели в году (от 1 до 53);

$\text{mod}$  – функция модуля числа;

$C_D$  – размер скидки в зависимости от дня недели.

Скидка по дням недели распределяется следующим образом:

Понедельник – 25%

Вторник – 22%

Среда – 19%

Четверг – 13%

Пятница – 24%

Суббота – 16%

Воскресенье – 18%

Скидка является целым числом и при вычислении всегда округляется в меньшую сторону. Определите, в какие дни 2024 года будет самая высокая скидка и чему она равна. В ответе укажите дату и размер скидки.

### Решение

Листинг программы, осуществляющий перебор всех дат.

Листинг на языке C++	Листинг на языке Python
<pre>#include &lt;iostream&gt; #include &lt;iomanip&gt; using namespace std;  int main() {     // день, месяц, счетчик дней     int D, M, dayNumber = 0;     // число дней в месяцах     int daysInMonth[12] = { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };     // день недели (0..6)     int weekNumber;     // скидка по дням недели     int weekDayDisc[7] = {         25, 22, 19, 13, 24, 16, 18 };     // общая скидка     double disc, discMax = 0;      // цикл по месяцам     for (M = 1; M &lt; 13; M++)     {         // цикл по дням в месяце         for (D=1; D&lt;daysInMonth[M-1]+1; D++)         {             // счетчик дней в году             dayNumber++;</pre>	<pre># число дней в месяцах days_in_month = [31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31] # скидка по дням недели weekDayDisc = [25, 22, 19, 13, 24, 16, 18]  # максимальная скидка discMax = 0 # день в году dayNumber = 0  # цикл по месяцам for M in range(1, 13):     # цикл по дням в месяце     for D in range(1, days_in_month[M-1] + 1):         # счетчик дней в году         dayNumber += 1         # номер недели в году         weekNumber = dayNumber // 7 + 1         # вычисление скидки по формуле         disc = int(((D % 10) * (M % 3) * weekDayDisc_v1[dayNumber % 7]) / (M + (abs(30 - weekNumber) + 1)))         # сравнение с сохраненной         # максимальной скидкой         # и перезапись, если</pre>

<pre>// счетчик недель в году weekNumber = dayNumber / 7 + 1; // вычисление скидки по формуле disc = ((D % 10) * (M % 3) * weekDayDisc[dayNumber%7]) / (M + (abs(30-weekNumber) + 1)); // сравнение с сохраненной // максимальной скидкой // и перезапись, если // текущая скидка больше if (disc &gt;= discMax) { discMax = disc; // вывод на экран даты и скидки cout &lt;&lt; "Discount:" &lt;&lt; disc &lt;&lt; " - Date: " &lt;&lt; D &lt;&lt; "." &lt;&lt; M &lt;&lt; " (" &lt;&lt; dayNumber &lt;&lt; ")" &lt;&lt; " dow: " &lt;&lt; (dayNumber % 7) + 1 &lt;&lt; endl; } } }</pre>	<pre># текущая скидка больше if disc &gt;= discMax: discMax = disc; # вывод на экран даты и скидки print(f"Discount: {disc} - Date: {D}.{M} ({dayNumber}) dow: {dayNumber%7+1}")</pre>
--	--

**Результат работы программы:**

```
Discount:0 - Date: 01.01 (1) dow: 2
Discount:1 - Date: 02.01 (2) dow: 3
Discount:1 - Date: 03.01 (3) dow: 4
Discount:3 - Date: 04.01 (4) dow: 5
Discount:3 - Date: 06.01 (6) dow: 7
Discount:5 - Date: 07.01 (7) dow: 1
Discount:5 - Date: 08.01 (8) dow: 2
Discount:5 - Date: 09.01 (9) dow: 3
Discount:6 - Date: 18.01 (18) dow: 5
Discount:7 - Date: 28.01 (28) dow: 1
Discount:7 - Date: 29.01 (29) dow: 2
Discount:7 - Date: 04.02 (35) dow: 1
Discount:8 - Date: 05.02 (36) dow: 2
Discount:8 - Date: 06.02 (37) dow: 3
Discount:14 - Date: 08.02 (39) dow: 5
Discount:16 - Date: 18.02 (49) dow: 1
Discount:18 - Date: 29.02 (60) dow: 5
Discount:25 - Date: 09.05 (130) dow: 5
Discount:30 - Date: 19.05 (140) dow: 1
Discount:34 - Date: 08.08 (221) dow: 5
```

**Ответ: Самая большая скидка будет 34% 08 августа 2024 года (пятница)**

**Вариант 2**

Магазин придумал свою систему скидок на весь 2024 год. Скидка (в процентах) вычисляется на основе даты по следующей формуле:

$$C = \frac{(D\%10) * (M\%3) * C_D}{M + (mod(30 - H) + 1)}$$

где

*D* – день месяца (от 1 до 28..31 в зависимости от месяца);

*M* – порядковый номер месяца (от 1 до 12);

$H$  – порядковый номер текущей недели в году (от 1 до 53);

$mod$  – функция модуля числа;

$C_d$  – размер скидки в зависимости от дня недели.

Скидка по дням недели распределяется следующим образом:

Понедельник – 23%

Вторник – 20%

Среда – 18%

Четверг – 20%

Пятница – 15%

Суббота – 17%

Воскресенье – 21%

Скидка является целым числом и при вычислении всегда округляется в меньшую сторону.

Определите, в какие дни 2024 года будет самая высокая скидка и чему она равна. В ответе укажите дату и размер скидки.

### Решение

Листинг программы, осуществляющий перебор всех дат.

Листинг на языке C++	Листинг на языке Python
<pre>#include &lt;iostream&gt; #include &lt;iomanip&gt; using namespace std;  int main() {     // день, месяц, счетчик дней     int D, M, dayNumber = 0;     // число дней в месяцах     int daysInMonth[12] = { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };     // день недели (0..6)     int weekNumber;     // скидка по дням недели     int weekDayDisc[7] = {         23, 20, 18, 20, 15, 17, 21 };     // общая скидка     double disc, discMax = 0;      // цикл по месяцам     for (M = 1; M &lt; 13; M++)     {         // цикл по дням в месяце         for (D=1; D&lt;daysInMonth[M-1]+1; D++)         {             // счетчик дней в году             dayNumber++;             // счетчик недель в году             weekNumber = dayNumber / 7 + 1;             // вычисление скидки по формуле             disc = ((D % 10) * (M % 3) *                 weekDayDisc[dayNumber%7]) /                 (M + (abs(30-weekNumber) + 1));             // сравнение с сохраненной             // максимальной скидкой             // и перезапись, если             // текущая скидка больше             if (disc &gt;= discMax)             {                 discMax = disc;             }             // вывод на экран даты и скидки</pre>	<pre># число дней в месяцах days_in_month = [31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31] # скидка по дням недели weekDayDisc = [23, 20, 18, 20, 15, 17, 21]  # максимальная скидка discMax = 0 # день в году dayNumber = 0  # цикл по месяцам for M in range(1, 13):     # цикл по дням в месяце     for D in range(1, days_in_month[M-1] + 1):         # счетчик дней в году         dayNumber += 1         # номер недели в году         weekNumber = dayNumber // 7 + 1         # вычисление скидки по формуле         disc = int(((D % 10) * (M % 3) * weekDayDisc_v1[dayNumber % 7]) / (M + (abs(30 - weekNumber) + 1)))         # сравнение с сохраненной         # максимальной скидкой         # и перезапись, если         # текущая скидка больше         if disc &gt;= discMax:             discMax = disc;             # вывод на экран даты и скидки             print(f"Discount: {disc} - Date: {D}.{M} ({dayNumber}) dow: {dayNumber%7+1}")</pre>

```
        cout << "Discount:" << disc <<
" - Date: " << D << "." << M << " (" <<
dayNumber << ")" << " dow: " <<
(dayNumber % 7) + 1 << endl;
    }
}
}
```

**Результат работы программы:**

Discount:0 - Date: 01.01 (1) dow: 2  
Discount:1 - Date: 02.01 (2) dow: 3  
Discount:1 - Date: 03.01 (3) dow: 4  
Discount:1 - Date: 04.01 (4) dow: 5  
Discount:2 - Date: 05.01 (5) dow: 6  
Discount:4 - Date: 06.01 (6) dow: 7  
Discount:5 - Date: 07.01 (7) dow: 1  
Discount:5 - Date: 08.01 (8) dow: 2  
Discount:5 - Date: 09.01 (9) dow: 3  
Discount:5 - Date: 19.01 (19) dow: 6  
Discount:5 - Date: 27.01 (27) dow: 7  
Discount:6 - Date: 28.01 (28) dow: 1  
Discount:6 - Date: 29.01 (29) dow: 2  
Discount:6 - Date: 04.02 (35) dow: 1  
Discount:7 - Date: 05.02 (36) dow: 2  
Discount:8 - Date: 06.02 (37) dow: 3  
Discount:10 - Date: 07.02 (38) dow: 4  
Discount:11 - Date: 09.02 (40) dow: 6  
Discount:11 - Date: 17.02 (48) dow: 7  
Discount:14 - Date: 18.02 (49) dow: 1  
Discount:14 - Date: 19.02 (50) dow: 2  
Discount:14 - Date: 06.05 (127) dow: 2  
Discount:14 - Date: 07.05 (128) dow: 3  
Discount:18 - Date: 08.05 (129) dow: 4  
Discount:21 - Date: 18.05 (139) dow: 7  
Discount:27 - Date: 19.05 (140) dow: 1  
Discount:27 - Date: 09.08 (222) dow: 6  
**Discount:28 - Date: 18.08 (231) dow: 1**

**Ответ: Самая большая скидка будет 28% 18 августа 2024 года (понедельник)**