

22 (повышенный уровень, время – 7 мин)

Тема: Построение математических моделей для решения практических задач. Архитектура современных компьютеров. Многопроцессорные системы

Что проверяется:

Умение анализировать алгоритм, содержащий ветвление и цикл

3.1.1. Программная и аппаратная организация компьютеров и компьютерных систем. Виды программного обеспечения.

1.3.2. Оценивать скорость передачи и обработки информации.

Что нужно знать:

- процессы в современных компьютерах могут выполняться параллельно, если являются независимыми
- выражение «процесс В зависит от процесса А» означает, что выполнение процесса В не может начаться раньше, чем выполнение процесса А

Пример задания:

Р-01 (демо-2024). В файле **22-2024.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Определите **максимальную продолжительность отрезка времени (в мс), в течение которого возможно одновременное выполнение четырёх процессов**, при условии, что все независимые друг от друга процессы могут выполняться параллельно.

Типовой пример организации данных в файле:

ID процесса В	Время выполнения процесса В (мс)	ID процесса(ов) А
1	4	0
2	3	0
3	1	1; 2
4	7	3

Решение (электронные таблицы):

1) используя таблицу в файле, построим диаграмму Ганта:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1	ID	Зависит от	Время		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
2	1	0	4	0	1	1	1	1																				
3	2	0	3	0	1	1	1																					
4	3	1; 2	5	0					1	1	1	1	1															
5	4	3	7	0											1	1	1	1	1	1	1							
6	5	3	6	0											1	1	1	1	1	1								
7	6	5	2	0																	1	1						
8	7	4; 6	5	0																			1	1	1	1	1	
9	8	7	2	0																							1	1
10	9	0	7	0	1	1	1	1	1	1	1																	
11	10	0	8	0	1	1	1	1	1	1	1	1																
12	11	9	6	0									1	1	1	1	1	1										
13	12	10	6	0										1	1	1	1	1	1									

Столбец D предназначен для контроля длительности процесса: длительность в столбце C должна совпадать с количеством единиц справа. Например, в ячейке D записана формула $=\text{СУММ}(E2:AB2) - C2$

Если все правильно, то в столбце D должны оказаться одни нули.

- 2) В каждой ячейке строки под диаграммой суммируем единицы в соответствующем столбце. Например, в ячейке E14 находится формула =СУММ(E2:E14):

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1	ID	Зависит от	Время		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
2	1	0	4	0	1	1	1	1																				
3	2	0	3	0	1	1	1																					
4	3	1; 2	5	0					1	1	1	1	1															
5	4	3	7	0										1	1	1	1	1	1	1								
6	5	3	6	0										1	1	1	1	1	1									
7	6	5	2	0																1	1							
8	7	4; 6	5	0																		1	1	1	1	1		
9	8	7	2	0																							1	1
10	9	0	7	0	1	1	1	1	1	1	1	1																
11	10	0	8	0	1	1	1	1	1	1	1	1	1															
12	11	9	6	0									1	1	1	1	1	1										
13	12	10	6	0										1	1	1	1	1	1									
14					4	4	4	3	3	3	3	3	3	4	4	4	4	3	2	2	1	1	1	1	1	1	1	1
15																												
16																												

Пока наибольший непрерывный временной отрезок, когда выполняются одновременно 4 процесса, длится 4 мс (он выделен синим цветом).

- 3) Однако, мы можем попробовать улучшить результат, передвигая процессы по временной оси, сохранив при этом все зависимости. Например, процессы 1, 2, 9 и 10, не зависящие от других процессов, могут начаться в любой момент. Процесс 3 может начаться в любой момент после окончания процессов 1 и 2 и т.д.
- 4) В частности, мы можем передвинуть на оси всю группу процессов 9-12 вправо так, чтобы процессы 9 и 10 начались на 10-й секунде. Тогда непрерывный временной отрезок, когда выполняются одновременно 4 процесса, длится 7 мс (он выделен синим цветом):

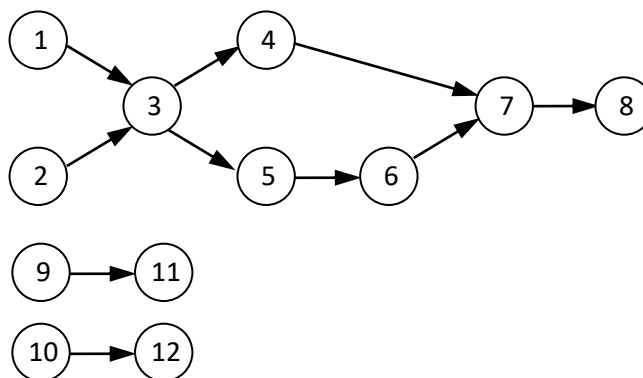
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
1	ID	Зависит от	Время		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
2	1	0	4	0	1	1	1	1																					
3	2	0	3	0	1	1	1																						
4	3	1; 2	5	0					1	1	1	1	1																
5	4	3	7	0										1	1	1	1	1	1	1									
6	5	3	6	0										1	1	1	1	1	1										
7	6	5	2	0																1	1								
8	7	4; 6	5	0																		1	1	1	1	1			
9	8	7	2	0																							1	1	
10	9	0	7	0										1	1	1	1	1	1	1									
11	10	0	8	0										1	1	1	1	1	1	1	1								
12	11	9	6	0																	1	1	1	1	1	1			
13	12	10	6	0																		1	1	1	1	1	1		
14					2	2	2	1	1	1	1	1	1	4	4	4	4	4	4	4	3	3	3	3	3	3	2	1	
15																													

Лучшего результата найти не удалось.

- 5) К сожалению, для решения этой задачи пока не придумали более систематического метода, который был бы как-то связан с курсом информатики в средней школе.
- 6) Ответ: **7**.

Решение (PRO100-ЕГЭ, <https://www.youtube.com/watch?v=nBD3LwQTGLg>):

- 1) построим граф, который показывает взаимосвязи процессов; у нас получится несколько цепочек:



- 2) из этой схемы очевидно, что при данных условиях максимальное количество одновременно выполняемых процессов – 4, причём должны быть задействованы две нижних цепочки (9-11 и 10-12), а также часть самой верхней схемы, где одновременно работают два процесса (1 и 2 или 4 и 5-6)
- 3) рассмотрим первый вариант: запустим сразу процессы 1, 2, 9 и 10:

	A	B	C	D	E	F	G	H	I	J
1	1	2	3	4	5	6	7	8	9	10
2	1	1	1	1						
3	2	2	2							
4	9	9	9	9	9	9	9			
5	10	10	10	10	10	10	10	10		
6										
7										

получается, что непрерывный отрезок, когда работают 4 процесса, длится всего 3 мс, потом процесс 2 заканчивается, а никакой другой процесс мы запустить не можем

- 4) рассмотрим второй вариант: дождёмся завершения процесса 3 и запустим одновременно процессы 4, 5, 9 и 10:

	A	B	C	D	E	F	G	H	I
1	1	2	3	4	5	6	7	8	9
2	4	4	4	4	4	4	4		
3	5	5	5	5	5	5			
4	9	9	9	9	9	9	9		
5	10	10	10	10	10	10	10	10	
6									
7									

получили, что непрерывный отрезок, во время которого работают 4 процесса, уже длится 6 мс

- 5) после окончания процесса 5 можно сразу запустить процесс 6 (продолжить цепочку):

	A	B	C	D	E	F	G	H
1	1	2	3	4	5	6	7	8
2	4	4	4	4	4	4	4	
3	5	5	5	5	5	5	6	6
4	9	9	9	9	9	9	9	
5	10	10	10	10	10	10	10	10
6								
7								

так удастся расширить нужный отрезок до 7 мс

- 6) чтобы «продлить» отрезок работы 4-х процессов, нужно продолжить цепочки после процессов 4 и 9 после 7 мс
- 7) в этот момент мы можем запустить только процесс 11 (после 9-го);

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	1	2	3	4	5	6	7	8	9	10	11	12	13
2	4	4	4	4	4	4	4						
3	5	5	5	5	5	5	6	6					
4	9	9	9	9	9	9	9	11	11	11	11	11	11
5	10	10	10	10	10	10	10	10					
6													
7													

- 8) запустить процесс 7 сразу после процесса 4 мы не можем, потому что ещё не закончился процесс 6; таким образом, не существует свободного процесса, который мог бы «продлить» процесс 4 после 7 мс
- 9) Ответ: **7**.

Ещё пример задания:

P-00 (демо-2023). В файле **22-0.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс B зависит от процесса A, если для выполнения процесса B необходимы результаты выполнения процесса A. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Определите минимальное время, через которое завершится выполнение всей совокупности процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.

Типовой пример организации данных в файле:

ID процесса B	Время выполнения процесса B (мс)	ID процесса(ов) A
1	4	0
2	3	0
3	1	1; 2
4	7	3

В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2, то есть, через 4 мс после старта. Он длится 1 мс и закончится через $4 + 1 = 5$ мс после старта. Выполнение процесса 4 может начаться только после завершения процесса 3, то есть, через 5 мс. Он длится 7 мс, так что минимальное время завершения всех процессов равно $5 + 7 = 12$ мс.

Решение (электронные таблицы):

- 1) таблица имеет вид:

ID процесса B	Время выполнения процесса B (мс)	ID процесса (ов) A
1	4	0
2	3	0
3	1	1; 2
4	7	3
5	6	3
6	3	5
7	1	4; 6
8	2	7
9	7	0

10	8	0
11	6	9
12	6	10

2) отсортируем данные в таблице так, чтобы:

- все независимые процессы оказались в начале таблицы
- любой процесс был расположен ПОСЛЕ всех процессов, от которых он зависит

ID процесса В	Время выполнения процесса В (мс)	ID процесса (ов) А
1	4	0
2	3	0
9	7	0
10	8	0
3	1	1; 2
4	7	3
5	6	3
6	3	5
7	1	4; 6
8	2	7
11	6	9
12	6	10

3) добавляем еще один столбец: время окончания процесса; для всех независимых процессов записываем в соответствующие ячейки время окончания, равное длительности процесса:

ID процесса В	Время выполнения процесса В (мс)	ID процесса (ов) А	Время окончания Т, мс
1	4	0	4
2	3	0	3
9	7	0	7
10	8	0	8
3	1	1; 2	
4	7	3	
5	6	3	
6	3	5	
7	1	4; 6	
8	2	7	
11	6	9	
12	6	10	

4) для остальных процессов определяем время окончания как сумму длительности этого процесса и максимального времени окончания процессов, от которых зависит данный процесс (здесь $T(x)$ обозначает время окончания процесса с идентификатором x):

$$T(3) = 1 + \max(T(1), T(2)) = 1 + 4 = 5$$

$$T(4) = 7 + T(3) = 7 + 5 = 12$$

$$T(5) = 6 + T(3) = 6 + 5 = 11$$

$$T(6) = 3 + T(5) = 3 + 11 = 14$$

$$T(7) = 1 + \max(T(4), T(6)) = 1 + 14 = 15$$

$$T(8) = 2 + T(7) = 2 + 15 = 17$$

$$T(11) = 6 + T(9) = 6 + 7 = 13$$

$$T(12) = 6 + T(10) = 6 + 8 = 14$$

ID процесса В	Время выполнения процесса В (мс)	ID процесса (ов) А	Время окончания Т, мс
1	4	0	4
2	3	0	3
9	7	0	7
10	8	0	8
3	1	1; 2	5
4	7	3	12
5	6	3	11
6	3	5	14
7	1	4; 6	15
8	2	7	17
11	6	9	13
12	6	10	14

5) время завершения совокупности всех процессов равно времени завершения последнего из них, поэтому нужно выбрать максимальное значение в последнем столбце

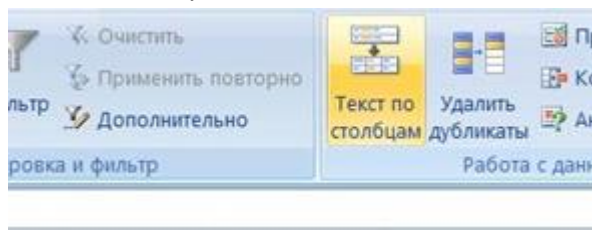
6) Ответ: **17**.

Решение (электронные таблицы, функция ВПР, Информатик БУ):

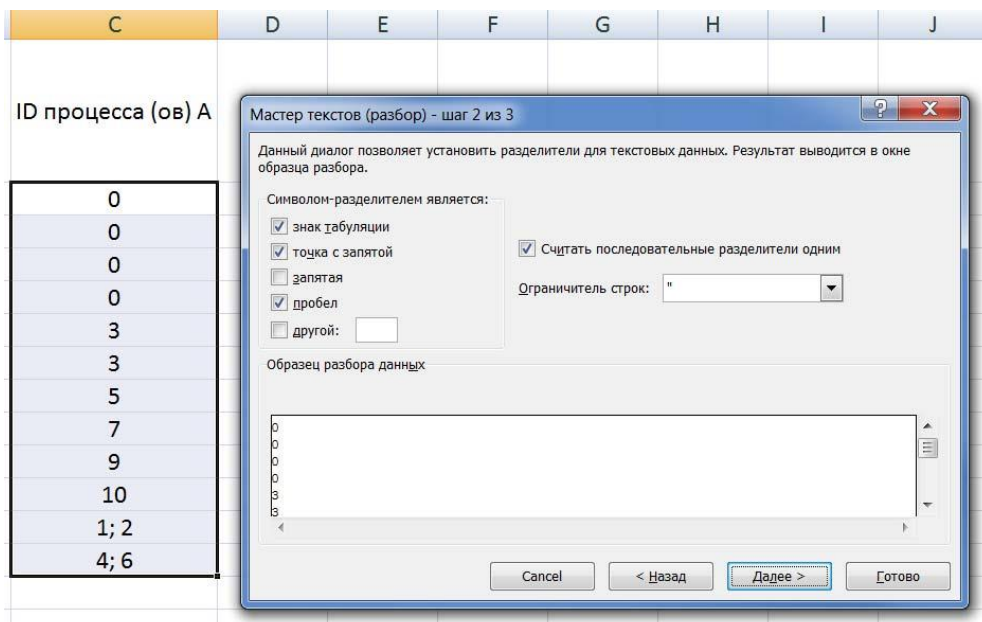
- Для начала отсортируем таблицу по столбцу С (**ID процесса (ов) А**). Это нужно для того, чтобы отдельно рассмотреть независимые процессы (те, у которых в столбце С находится значение 0); вместо этого можно было бы использовать функцию ЕСЛИ, но в этом случае формула была бы немного сложнее;

	A	B	C
	ID процесса В	Время выполнения процесса В (мс)	ID процесса (ов) А
1			
2	1	4	0
3	2	3	0
4	9	7	0
5	10	8	0
6	4	7	3
7	5	6	3
8	6	3	5
9	8	2	7
10	11	6	9
11	12	6	10
12	3	1	1; 2
13	7	1	4; 6
14			

- Теперь необходимо расцепить процессы в ячейках столбца С, в которых находится более одного процесса; в данном примере это можно было бы сделать вручную, но ручной способ не подойдет для больших таблиц. Для этого выделяем нужные ячейки столбца С, переходим в меню «Данные», и нажимаем кнопку «Текст по столбцам»:



В появившемся окне выбираем формат данных «с разделителями», нажимаем кнопку «Далее», и указываем символы, которые разделяют процессы в ячейках. В нашем случае это точка с запятой и пробел. Нажимаем кнопку «Готово».



После этого один из процессов останется в столбце C, а второй отправится в столбец D (если процессов больше, чем 2, номера оставшихся размещаются в столбцах E, F и далее).

- 3) Посчитаем в столбце E полное время выполнения каждого процесса (время самого процесса + время процессов, связанных с ним). Сначала отдельно рассмотрим независимые процессы. Для этого в ячейке E2 пишем формулу `=B2`, и копируем её для всех независимых процессов.

	A	B	C	D	E
	ID процесса B	Время выполнения процесса B (мс)	ID процесса (ов) A		
1					
2	1	4	0		4
3	2	3	0		3
4	9	7	0		7
5	10	8	0		<code>=B5</code>
6	4	7	3		

- 4) Теперь рассмотрим процессы B, которые зависят только от одного процесса. Для этого возьмем время выполнения самого процесса B, и прибавим к нему полное время выполнения процесса A (ID процесса A находится в столбце C, а полное время выполнения – в столбце E). Для поиска времени выполнения процесса A воспользуемся функцией ВПР. В ячейку E6 запишем формулу:

`=B6+ВПР(С6;А:Е;5;0)`

и скопируем её для всех процессов B, у которых указан только один процесс A.

	A	B	C	D	E	F
	ID процесса B	Время выполнения процесса B (мс)	ID процесса (ов) A			
1						
2	1	4	0		4	
3	2	3	0		3	
4	9	7	0		7	
5	10	8	0		8	
6	4	7	3		7	
7	5	6	3		6	
8	6	3	5		9	
9	8	2	7		2	
10	11	6	9		13	
11	12	6	10		<code>=B11+ВПР(С11;А:Е;5;0)</code>	
12	3	1	1	2		
13	7	1	4	6		
14						

- 5) Далее рассмотрим процессы В, которые зависят уже от двух процессов. Так как процессы в столбцах С и D могут выполняться параллельно, будем брать максимальное значение среди времени их выполнения, и прибавлять к этому значению время выполнения самого процесса В. Для этого в ячейку E12 запишем формулу:

=B12+МАКС(ВПР(С12;А:Е;5;0);ВПР(D12;А:Е;5;0))

и копируем её для всех процессов В, у которых указано два процесса А.

	A	B	C	D	E	F	G	H	I
	ID процесса В	Время выполнения процесса В (мс)	ID процесса (ов) А						
1									
2	1	4	0		4				
3	2	3	0		3				
4	9	7	0		7				
5	10	8	0		8				
6	4	7	3		12				
7	5	6	3		11				
8	6	3	5		14				
9	8	2	7		17				
10	11	6	9		13				
11	12	6	10		14				
12	3	1	1	2	5				
13	7	1	4		=B13+МАКС(ВПР(C13;А:Е;5;0);ВПР(D13;А:Е;5;0))				
14									
15									

- 6) Так мы получили столбец Е, в ячейках которого указано полное время выполнения каждого процесса В. Так как процессы могут выполняться параллельно, нам остаётся найти максимальное значение в ячейках столбца Е. Для этого в любую пустую ячейку пишем формулу: **=МАКС(А:А)** и получаем ответ: 17.
- 7) Ответ: **17**.

Решение (электронная таблица, С. Кох):

- 1) сортируем таблицу по первому столбцу (ID процесса В) и добавляем первой строкой (в электронной таблице – в строку 2) фиктивный процесс в которую записываем процесс с ID = 0, время выполнения 0 и время окончания процесса 0:

	A	B	C
	ID процесса В	Время выполнения процесса В (мс)	ID процесса (ов) А
1			
2	0	0	0
3	1	4	0
4	2	3	0
5	3	1	1; 2
6	4	7	3
7	5	6	3
8	6	3	5
9	7	1	4; 6
10	8	2	7
11	9	7	0
12	10	8	0
13	11	6	9
14	12	6	10

- 2) расцепляем процессы в ячейках столбца С с помощью функции «Текст по столбцам» (см. выше в решении Информатика БУ):

	A	B	C	D
	ID процесса B	Время выполнения процесса B (мс)	ID процесса (ов) A	
1				
2	0	0	0	
3	1	4	0	
4	2	3	0	
5	3	1	1	2
6	4	7	3	
7	5	6	3	
8	6	3	5	
9	7	1	4	6
10	8	2	7	
11	9	7	0	
12	10	8	0	
13	11	6	9	
14	12	6	10	

- 3) В первом свободном столбце в 3-й строке пишем формулу, которая учитывает максимальное количество процессов, которое встречается в этой задаче (например, здесь в E3 пишем формулу =МАКС(ВПР(C3;A:E;5);ВПР(D3;A:E;5))+B3) и растягиваем эту формулу на весь столбец. В этом случае все пустые ячейки Excel считает равными 0 и обращается к 0 процессу, время завершения которого 0.

	A	B	C	D	E	F	G	H
	ID процесса B	Время выполнения процесса B (мс)	ID процесса (ов) A					
1								
2	0	0	0					
3	1	4	0		=МАКС(ВПР(C3;A:E;5);ВПР(D3;A:E;5))+B3			
4	2	3	0		3			
5	3	1	1	2	5			
6	4	7	3		12			
7	5	6	3		11			
8	6	3	5		14			
9	7	1	4	6	15			
10	8	2	7		17			
11	9	7	0		7			
12	10	8	0		8			
13	11	6	9		13			
14	12	6	10		14			

- 4) находим максимум в последнем столбце:

F3						=МАКС(E:E)
	A	B	C	D	E	F
	ID процесса B	Время выполнения процесса B (мс)	ID процесса (ов) A			
1						
2	0	0	0			
3	1	4	0		4	17
4	2	3	0		3	
5	3	1	1	2	5	
6	4	7	3		12	
7	5	6	3		11	
8	6	3	5		14	
9	7	1	4	6	15	
10	8	2	7		17	
11	9	7	0		7	
12	10	8	0		8	
13	11	6	9		13	
14	12	6	10		14	

5) Ответ: **17**.

Решение (диаграмма Ганта, А. Кожевникова):

- Решение удобно выполнять в Excel, построив диаграмму Ганта, которая показывает зависимость процессов. Построение диаграммы состоит в закрашивании ячеек таблицы Excel разным цветом, длина (количество закрашиваемых ячеек) процесса соответствует времени его выполнения.
- Построим диаграмму Ганта для таблицы, приведённой в задании:

ID процесса B	Время выполнения процесса B (мс)	ID процесса (ов) A
1	4	0
2	3	0
3	1	1; 2
4	7	3
5	6	3
6	3	5
7	1	4; 6
8	2	7
9	7	0
10	8	0
11	6	9
12	6	10

- Сначала отобразим на диаграмме процесс 1 и 2. Оба процесса начинаются независимо друг от друга. Поэтому строим оба процесса с самого начала.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1														
2	2	2	2														

- процесс 3 длительностью 1 начинается только после окончания процессов 1 и 2:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	3													
2	2	2	2														

- 5) Процессы 4 и 5 начинаются после выполнения процесса 3:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1				3	4												
2					5												

- 6) Процесс 6 начинается после выполнения процесса 5. Процесс 7 начинается после завершения процессов 4 и 6. Процесс 8 начинается после завершения процесса 7.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1				3	4										7	8	
2					5							6					

- 7) Процессы 9 и 10 начинаются сразу в момент 0, так как независимы от других процессов. Процесс 11 начинается после завершения процесса 9, а процесс 12 процесс сразу после завершения процесса 10.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1				3	4										7	8	
2					5							6					
9						11											
10								12									

- 8) Позже всех – через 17 мс – заканчивается процесс 8. Это и есть минимальное время завершения всей совокупности процессов.
- 9) Ответ: **17**.

Решение (программа):

- 1) можно решить задачу с помощью программы, если сохранить файл в CSV-формате (меню *Файл – Сохранить как*); файл будет выглядеть так:

```
ID процесса В;Время выполнения процесса В (мс);ID процесса (ов) А;
1;4;0;
2;3;0;
3;1;"1; 2";
4;7;3;
5;6;3;
6;3;5;
7;1;"4; 6";
8;2;7;
9;7;0;
10;8;0;
11;6;9;
12;6;10;
```

- 2) для чтения CSV-файла проще всего применить объект **reader** из стандартного модуля **csv**:

```
from csv import reader
with open( "22-0.csv", encoding="cp1251" ) as F:
    rdr = reader( F, delimiter=';', quotechar='"' ) # (1)
    next( rdr ) # читаем заголовки и пропускаем их # (2)
    ...
```

параметр **encoding**, переданный функции **open**, означает, что файл сохранен в кодировке Windows (CP-1251), именно так сохраняет CSV-файлы Excel; тут проблема только в чтении русских букв, поэтому можно просто заранее удалить из файла первую строку с заголовками

- 3) в строке (1) создается объект `reader`, в параметрах указываем, что разделитель данных (**`delimiter`**) – точка с запятой, а знак " используется как кавычка (**`quotechar`**)
- 4) в строке (2) мы читаем (и не сохраняем) строку с заголовками, если, конечно, не удалили ее заранее; если удалили, то функцию **`next()`** вызывать не нужно
- 5) теперь займемся хранением данных; будем использовать два словаря:
 - словарь **`finalTime`** содержит уже известные значения времени окончания процессов (вызов: **`finalTime[id]`** даёт время окончания процесса с идентификатором **`id`**)
 - словарь **`data`** содержит данные тех процессов, для которых время окончания неизвестно: вызов **`data[id]`** даёт кортеж **`(t, dependsOn)`**, где
 - `t`** – длительность процесса с идентификатором **`id`**,
 - `dependsOn`** – массив идентификаторов процессов, от которых зависит данный процесс

- 6) в цикле читаем данные из файла;

```
for s in rdr:
    id, t = int(s[0]), int(s[1])
    dependsOn = list( map( int, s[2].split(';') ) )
    if dependsOn == [0]:
        finalTime[id] = t # известно время окончания
    else:
        data[id] = ( t, dependsOn ) # неизвестно время окончания
```

массив строк **`s`** содержит данные одной строки CSV-файла

- **`s[0]`** – идентификатор (эту строку нужно преобразовать в целое число, вызвав **`int`**)
- **`s[1]`** – длительность процесса (тоже нужно преобразовать в целое число)
- **`s[2]`** – перечисление идентификаторов процессов, от которых зависит данный процесс; эта строка может иметь, например, такой вид:

```
'1; 2'
```

в результате вызова **`s[2].split(';')`** получаем такой массив из двух строк

```
['1', '2']
```

применяя к каждому элементу функцию **`int`**, получаем массив целых чисел **`dependsOn`**:

```
[1, 2]
```

Если этот массив состоит только из нуля, сразу записываем в словарь **`finalTime`** идентификатор и соответствующее ему время окончания процесса, которое равно длительности процесса (считаем, что он стартует в момент 0):

```
if dependsOn == [0]:
    finalTime[id] = t # известно время окончания
else:
    data[id] = ( t, dependsOn ) # неизвестно время окончания
```

Если же у процесса есть зависимости, записываем в словарь **`data`** кортеж, в котором хранятся длительность процесса и массив **`dependsOn`** идентификаторов процессов, от которых он зависит.

- 7) самое интересное – это обработка данных словаря **`data`**; все эти данные нужно преобразовать в соответствующие элементы словаря **`finalTime`** так чтобы словарь **`data`** стал пустым; делаем так:

```
while data:
    ids = list(data.keys())
    for id in ids:
        # если известны моменты окончания всех процессов, от
        # которых зависит процесс id, записать в finalTime время
        # окончания этого процесса и удалить данные процесса id из
        # словаря data
```

- 8) чтобы определить, что моменты окончания всех нужных процессов известны, используем функцию `all`:
- ```
if all((x in finalTime) for x in dependsOn):
 ...
```
- это значит «если все идентификаторы из массива `dependsOn` уже есть в словаре `finalTime`»
- 9) определяем время старта процесса `id` как максимальное время завершения всех процессов, от которых он зависит:
- ```
startId = max( finalTime[x] for x in dependsOn )
```
- 10) вычисляем время окончания процесса `id`:
- ```
finalTime[id] = startId + t
```
- и удаляем данные этого процесса из массива `data`:
- ```
del data[id]
```
- 11) вот полный цикл обработки:
- ```
while data:
 ids = list(data.keys())
 for id in ids:
 t, dependsOn = data[id]
 if all((x in finalTime) for x in dependsOn):
 startId = max(finalTime[x] for x in dependsOn)
 finalTime[id] = startId + t
 del data[id]
```
- 12) ответ – это максимальное время окончания процесса из словаря `finalTime`:
- ```
print( "Ответ:", max(finalTime.values()) )
```
- 13) полная программа:
- ```
from csv import reader
with open("22-0.csv", encoding="cp1251") as F:
 rdr = reader(F, delimiter=';', quotechar='"')
 next(rdr) # читаем заголовки и пропускаем их
 data = {}
 finalTime = {}
 for s in rdr:
 id, t = int(s[0]), int(s[1])
 dependsOn = list(map(int, s[2].split(';')))
 if dependsOn == [0]:
 finalTime[id] = t # известно время окончания
 else:
 data[id] = (t, dependsOn) # неизвестно время окончания
 while data:
 ids = list(data.keys())
 for id in ids:
 t, dependsOn = data[id]
 if all((x in finalTime) for x in dependsOn):
 startId = max(finalTime[x] for x in dependsOn)
 finalTime[id] = startId + t
 del data[id]
 print("Ответ:", max(finalTime.values()))
```
- 14) обратим внимание, что программа не использует никаких предположений о расположении данных о процессах в исходном списке, они могут быть расположены в произвольном порядке

- 15) программа заикнется, если данные будут некорректны, например, процесс 1 зависит от процесса 2, а процесс 2 – от процесса 1 (циклическая ссылка)
- 16) Ответ: 17.

### Задачи для тренировки:

- 1) (В. Шубинкин) В файле **22-1.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Определите минимальное время, через которое завершится выполнение всей совокупности процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |

В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2, то есть, через 4 мс после старта. Он длится 1 мс и закончится через  $4 + 1 = 5$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса 3, то есть, через 5 мс. Он длится 7 мс, так что минимальное время завершения всех процессов равно  $5 + 7 = 12$  мс.

- 2) (В. Шубинкин) В файле **22-2.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 3) (В. Шубинкин) В файле **22-3.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 4) (В. Шубинкин) В файле **22-4.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 5) (А. Кожевникова) В файле **22-5.xls** содержится информация о процессах внутри проектов P1 и P2. Каждый проект состоит из совокупности вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты процесса А. В этом случае процессы могут выполняться только последовательно. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы — время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Найдите разницу между минимальным временем выполнения проектов P1 и P2. Проект считается завершенным, когда завершились все процессы проекта.

- 6) **(А. Кожевникова)** В файле **22-5.xls** содержится информация о процессах внутри проектов P1 и P2. Каждый проект состоит из совокупности вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты процесса А. В этом случае процессы могут выполняться только последовательно. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы — время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0.  
Найдите минимальное время завершения процесса 12 из проекта P1.
- 7) **(А. Кожевникова)** В файле **22-5.xls** содержится информация о процессах внутри проектов P1 и P2. Каждый проект состоит из совокупности вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты процесса А. В этом случае процессы могут выполняться только последовательно. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы — время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0.  
Найдите минимальное время завершения процесса 4 из проекта P2.
- 8) **(Л. Шастин)** В файле **22-6.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).  
Среди всех независимых процессов найдите самый длительный и самый быстрый (заканчивающийся за минимальное время). В качестве ответа укажите разницу между временами выполнения этих процессов.
- 9) **(Л. Шастин)** В файле **22-6.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).  
Эта группа процессов выполняется дважды при различных условиях:  
1) все процессы считаются независимыми и выполняются параллельно;  
2) независимые процессы выполняются параллельно, а зависимые — последовательно.  
Определите, на сколько миллисекунд один режим быстрее другого.
- 10) **(Л. Евич)** В файле **22-1e.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 11) **(Л. Евич)** В файле **22-2e.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 12) **(Л. Евич)** В файле **22-3e.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 13) **(Л. Евич)** В файле **22-4e.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 14) **(Л. Евич)** В файле **22-5e.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).



- 15) (Л. Евич) В файле **22-6e.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 16) (Л. Евич) В файле **22-7e.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 17) (Л. Евич) В файле **22-8e.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 18) (Л. Евич) В файле **22-9e.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 19) (Л. Евич) В файле **22-10e.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 20) (Л. Евич) В файле **22-11e.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 21) (Л. Евич) В файле **22-12e.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 22) (Л. Евич) В файле **22-13e.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 23) (Л. Евич) В файле **22-14e.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 24) (PRO100 ЕГЭ) В файле **22-24.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 25) (PRO100 ЕГЭ) В файле **22-25.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 26) (PRO100 ЕГЭ) В файле **22-26.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 27) (PRO100 ЕГЭ) В файле **22-27.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 28) (PRO100 ЕГЭ) В файле **22-28.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 29) (PRO100 ЕГЭ) В файле **22-29.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В требуется, чтобы был выполнен **хотя бы один из процессов А**. Информация о процессах представлена в файле в виде таблицы. В первой строке таблицы указан идентификатор процесса (ID), во второй строке таблицы – время его выполнения в миллисекундах, в третьей строке перечислены с разделителем «;» ID процессов,



от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Определите минимальное время, через которое завершится выполнение всей совокупности процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |

В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения процесса 1 или 2, то есть, через 3 мс после старта. Он длится 1 мс и закончится через  $3 + 1 = 4$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса 3, то есть, через 4 мс. Он длится 7 мс, так что минимальное время завершения всех процессов равно  $4 + 7 = 11$  мс.

- 30) (PRO100 ЕГЭ) В файле **22-30.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В требуется, чтобы был выполнен **хотя бы 50% от количества процессов А**. Информация о процессах представлена в файле в виде таблицы. В первой строке таблицы указан идентификатор процесса (ID), во второй строке таблицы – время его выполнения в миллисекундах, в третьей строке перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Определите минимальное время, через которое завершится выполнение всей совокупности процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 1; 2; 3           |

В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения **хотя бы 50% из двух процессов**, то есть любого процесса 1 или 2, то есть, через 3 мс после старта. Он длится 1 мс и закончится через  $3 + 1 = 4$  мс после старта. Выполнение процесса 4 может начаться только после завершения **хотя бы 50% из трёх процессов**, то есть любых двух из процессов (1, 2, 3), то есть, через 4 мс. Он длится 7 мс, так что минимальное время завершения всех процессов равно  $4 + 7 = 11$  мс.

- 31) (А. Носкин) В файле **22-31.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID

процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. При составлении таблицы была потеряна информация о том, после какого процесса А начался процесс В с ID = 12. Однако известно, что вся совокупности процессов завершилась за минимальное время 25 мс. Определите ID процесса, после которого начался процесс с ID = 12. В ответе укажите только число.

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | ?                 |

В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2, то есть, через 4 мс после старта. Он длится 1 мс и закончится через  $4 + 1 = 5$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса, ID которого **потеряно**. Его продолжительность равно 7 мс. Если бы минимальное время завершения всех процессов была равно 12 мс, то процесс 4 начинался бы **после процесса 3** ( $12 - 7 = 5$  мс).

- 32) (А. Кабанов) В файле **22-32.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Определите максимально возможное целочисленное неизвестное время выполнения процесса t, при котором выполнение всей совокупности процессов может завершиться не более чем за 134 мс.

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | t                                | 1; 2              |
| 4             | 7                                | 3                 |

Пусть выполнение данной совокупности процессов закончилось за 15 мс. В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2, то есть, через 4 мс после старта. Он длится t мс и закончится через  $4 + t$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса 3, то есть, через  $4 + t$  мс. Он длится 7 мс, так что минимальное время завершения всех процессов равно  $4 + t + 7 = 15$  мс. Следовательно,  $t = 15 - 4 - 7 = 4$  мс. Ответ для этого примера: 4.

- 33) (А. Кабанов) В файле **22-33.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы

указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Определите максимально возможное целочисленное неизвестное время выполнения процесса  $t$ , при котором выполнение всей совокупности процессов может завершиться не более чем за 107 мс.

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | $t$                              | 1; 2              |
| 4             | 7                                | 3                 |

Пусть выполнение данной совокупности процессов закончилось за 15 мс. В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2, то есть, через 4 мс после старта. Он длится  $t$  мс и закончится через  $4 + t$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса 3, то есть, через  $4 + t$  мс. Он длится 7 мс, так что минимальное время завершения всех процессов равно  $4 + t + 7 = 15$  мс. Следовательно,  $t = 15 - 4 - 7 = 4$  мс. Ответ для этого примера: 4.

- 34) (А. Кабанов) В файле **22-34.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Определите максимально возможное целочисленное неизвестное время выполнения процесса  $t$ , при котором выполнение всей совокупности процессов может завершиться не более чем за 96 мс. Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | $t$                              | 1; 2              |
| 4             | 7                                | 3                 |

Пусть выполнение данной совокупности процессов закончилось за 15 мс. В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2, то есть, через 4 мс после старта. Он длится  $t$  мс и закончится через  $4 + t$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса 3, то есть, через  $4 + t$  мс. Он длится 7 мс, так что минимальное время завершения всех процессов равно  $4 + t + 7 = 15$  мс. Следовательно,  $t = 15 - 4 - 7 = 4$  мс. Ответ для этого примера: 4.

- 35) (А. Кабанов) В файле **22-35.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы

указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. При составлении таблицы была потеряна информация о том, после какого процесса начался процесс ID = 16. Однако известно, что вся совокупности процессов завершилась за минимальное время 138 мс. Определите ID процесса, после которого начался процесс с ID = 16. В ответе укажите только число.

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | ?                 |

В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2, то есть, через 4 мс после старта. Он длится 1 мс и закончится через  $4 + 1 = 5$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса, ID которого **потеряно**. Его продолжительность равно 7 мс. Если бы минимальное время завершения всех процессов была равно 12 мс, то процесс 4 начинался бы **после процесса 3** ( $12 - 7 = 5$  мс).

- 36) (А. Кабанов) В файле **22-36.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. При составлении таблицы была потеряна информация о том, после какого процесса начался процесс ID = 12. Однако известно, что вся совокупности процессов завершилась за минимальное время 154 мс. Определите ID процесса, после которого начался процесс с ID = 12. В ответе укажите только число.

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | ?                 |

В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2, то есть, через 4 мс после старта. Он длится 1 мс и закончится через  $4 + 1 = 5$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса, ID которого **потеряно**. Его продолжительность равно 7 мс. Если бы минимальное время завершения всех процессов была равно 12 мс, то процесс 4 начинался бы **после процесса 3** ( $12 - 7 = 5$  мс).

- 37) (А. Кабанов) В файле **22-37.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. При составлении таблицы была потеряна информация о том, после какого процесса начался процесс ID = 18. Однако известно, что вся совокупности процессов завершилась за минимальное время 158 мс. Определите ID процесса, после которого начался процесс с ID = 18. В ответе укажите только число.

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | ?                 |

В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2, то есть, через 4 мс после старта. Он длится 1 мс и закончится через  $4 + 1 = 5$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса, ID которого **потеряно**. Его продолжительность равно 7 мс. Если бы минимальное время завершения всех процессов была равно 12 мс, то процесс 4 начинался бы **после процесса 3** ( $12 - 7 = 5$  мс).

- 38) (А. Кабанов) В файле **22-38.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. **Если процесс В зависит от процесса А, то процесс В может начать выполнение не раньше, чем через 5 мс после завершения процесса А.** Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0.

Определите, за какое минимальное время можно выполнить все процессы. В ответе запишите целое число – минимальное время в мс.

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |

В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2 и 5 мс ожидания, то есть, через 9 мс после старта. Он длится 1 мс и закончится через  $9 + 1 = 10$  мс после старта. Выполнение процесса 4 может

начаться только после завершения процесса 3 и 5 мс ожидания, то есть, через 15 мс. Он длится 7 мс, так что минимальное время завершения всех процессов равно  $15 + 7 = 22$  мс.

- 39) (А. Кабанов) В файле **22-39.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс  $B$  зависит от процесса  $A$ , если для выполнения процесса  $B$  необходимы результаты выполнения процесса  $A$ . В этом случае процессы могут выполняться только последовательно. **Если процесс  $B$  зависит от процесса  $A$ , то процесс  $B$  может начать выполнение не раньше, чем через 7 мс после завершения процесса  $A$ .** Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0.

Определите, за какое минимальное время можно выполнить все процессы. В ответе запишите целое число – минимальное время в мс.

Типовой пример организации данных в файле:

| ID процесса B | Время выполнения процесса B (мс) | ID процесса(ов) A |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |

В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2 и 7 мс ожидания, то есть, через 11 мс после старта. Он длится 1 мс и закончится через  $11 + 1 = 12$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса 3 и 7 мс ожидания, то есть, через 19 мс. Он длится 7 мс, так что минимальное время завершения всех процессов равно  $19 + 7 = 26$  мс.

- 40) (А. Кабанов) В файле **22-40.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс  $B$  зависит от процесса  $A$ , если для выполнения процесса  $B$  необходимы результаты выполнения процесса  $A$ . В этом случае процессы могут выполняться только последовательно. **Если процесс  $B$  зависит от процесса  $A$ , то процесс  $B$  может начать выполнение не раньше, чем через 3 мс после завершения процесса  $A$ .** Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0.

Определите, за какое минимальное время можно выполнить все процессы. В ответе запишите целое число – минимальное время в мс.

Типовой пример организации данных в файле:

| ID процесса B | Время выполнения процесса B (мс) | ID процесса(ов) A |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |

В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2 и 3 мс ожидания, то есть, через 7 мс после старта.



Он длится 1 мс и закончится через  $7 + 1 = 8$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса 3 и 3 мс ожидания, то есть, через 11 мс. Он длится 7 мс, так что минимальное время завершения всех процессов равно  $11 + 7 = 18$  мс.

- 41) (А. Кабанов) В файле **22-41.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 42) (А. Кабанов) В файле **22-42.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 43) (А. Кабанов) В файле **22-43.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 44) (А. Кабанов) В файле **22-44.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс  $B$  зависит от процесса  $A$ , если для выполнения процесса  $B$  необходимы результаты выполнения процесса  $A$ . В этом случае процессы могут выполняться только последовательно. **Если процесс  $B$  зависит от процесса  $A$ , то процесс  $B$  может начать выполнение не раньше, чем через 9 мс после завершения процесса  $A$ .** Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0.

Определите, за какое минимальное время можно выполнить все процессы. В ответе запишите целое число – минимальное время в мс.

Типовой пример организации данных в файле:

| ID процесса B | Время выполнения процесса B (мс) | ID процесса(ов) A |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |

В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2 и 9 мс ожидания, то есть, через 13 мс после старта. Он длится 1 мс и закончится через  $13 + 1 = 14$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса 3 и 9 мс ожидания, то есть, через 23 мс. Он длится 7 мс, так что минимальное время завершения всех процессов равно  $23 + 7 = 30$  мс.

- 45) (Д. Статный) В файле **22-45.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс  $B$  зависит от процесса  $A$ , если для выполнения процесса  $B$  необходимы результаты выполнения процесса  $A$ . В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Определите **максимальное количество процессов, которые выполнялись параллельно**, при условии, что все независимые друг от друга процессы стартовали одновременно, а зависимые процессы стартовали одновременно с завершением всех процессов, от которых они зависят. Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |

В данном случае независимые процессы 1 и 2 могут выполняться параллельно с самого начала, остальные процессы выполняются последовательно. Ответ - 2.

- 46) (А. Куканова) В файле **22-46.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Известно, что независимые процессы начали выполняться сразу, остальные — как только их выполнение стало возможно. **Определите наибольшее количество процессов, выполнявшихся параллельно.** В ответе укажите **сумму этого количества и времени, прошедшего с момента начала вычислений до их завершения.**

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |

В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2, то есть, через 4 мс после старта. Он длится 1 мс и закончится через  $4 + 1 = 5$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса 3, то есть, через 5 мс. Он длится 7 мс, так что минимальное время завершения всех процессов равно  $5 + 7 = 12$  мс. При этом параллельно выполнялось не более 2 процессов (ID 1 и 2). Ответом для этого примера будет сумма 2 и 12, т.е. 14.

- 47) (А. Куканова) В файле **22-46.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Вычислительное устройство имеет **4 вычислительных ядра**. Каждый из параллельных процессов выполняется на отдельном ядре. **Из готовых к выполнению процессов в первую очередь запускаются процессы с наименьшими ID. Определите минимальное время, через которое завершится выполнение всей совокупности процессов.**

Типовой пример организации данных в файле:



| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |
| 5             | 5                                | 0                 |

Рассмотрим пример выше в случае, когда устройство имеет 2 вычислительных ядра: ядро I и ядро II. Независимые процессы 1, 2 и 5 готовы к выполнению, но запустятся только процессы 1 (пусть на ядре I) и 2 (на ядре II) как имеющие меньшие значения ID. При этом процесс 2 завершится через 3 мс. Поскольку процесс 3 может начаться только после завершения обоих процессов 1 и 2, на освободившемся ядре II запускается процесс 5, который завершится через  $3 + 5 = 8$  мс после старта. Процесс 1 завершится через 4 мс после старта и позволит начать выполнение процесса 3 на освободившемся ядре I. Процесс 3 завершится через  $4 + 1 = 5$  мс после старта. На освободившемся ядре I начнётся выполнение процесса 4, которое продлится 7 мс и закончится через  $5 + 7 = 12$  мс после начала вычислений. Таким образом, минимальное время завершения всех процессов равно 12 мс. Ответ: 12.

- 48) (А. Куканова) В файле **22-46.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Вычислительное устройство имеет **3 вычислительных ядра**. Каждый из параллельных процессов выполняется на отдельном ядре. **Из готовых к выполнению процессов в первую очередь запускаются процессы с наименьшими ID. Определите минимальное время, через которое завершится выполнение всей совокупности процессов.**

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |
| 5             | 5                                | 0                 |

Рассмотрим пример выше в случае, когда устройство имеет 2 вычислительных ядра: ядро I и ядро II. Независимые процессы 1, 2 и 5 готовы к выполнению, но запустятся только процессы 1 (пусть на ядре I) и 2 (на ядре II) как имеющие меньшие значения ID. При этом процесс 2 завершится через 3 мс. Поскольку процесс 3 может начаться только после завершения обоих процессов 1 и 2, на освободившемся ядре II запускается процесс 5, который завершится через  $3 + 5 = 8$  мс после старта. Процесс 1 завершится через 4 мс после старта и позволит начать выполнение процесса 3 на освободившемся ядре I. Процесс 3 завершится через  $4 + 1 = 5$  мс после старта. На освободившемся ядре I начнётся выполнение процесса 4, которое продлится 7 мс и закончится через  $5 + 7 = 12$  мс после начала вычислений. Таким образом, минимальное время завершения всех процессов равно 12 мс. Ответ: 12.

- 49) (А. Куканова) В файле **22-46.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Вычислительное устройство имеет **4 вычислительных ядра**. Каждый из параллельных процессов выполняется на отдельном ядре. Будем говорить, что ядро простаивает, если ядро не выполняет никакой процесс и при этом не вся совокупность процессов выполнена. Из готовых к выполнению процессов **в первую очередь запускаются процессы с наименьшим временем выполнения** (если таких больше, чем свободных ядер, приоритет имеют процессы с меньшими ID). Известно, что выполнение всей совокупности процессов заняло минимальное возможное время. **Определите суммарное время простоя всех ядер.**

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |
| 5             | 5                                | 0                 |

Рассмотрим пример выше в случае, если устройство имеет 2 вычислительных ядра: ядро I и ядро II. Независимые процессы 1, 2 и 5 готовы к выполнению, но запустятся только процессы 1 (пусть на ядре I) и 2 (на ядре II) как имеющие меньшее время выполнения. При этом процесс 2 завершится через 3 мс. Поскольку процесс 3 может начаться только после завершения обоих процессов 1 и 2, на освободившемся ядре II запускается процесс 5, который завершится через  $3 + 5 = 8$  мс после старта. Процесс 1 завершится через 4 мс после старта и позволит начать выполнение процесса 3 на освободившемся ядре I. Процесс 3 завершится через  $4 + 1 = 5$  мс после старта. На освободившемся ядре I начнётся выполнение процесса 4, которое продлится 7 мс и закончится через  $5 + 7 = 12$  мс после начала вычислений. Таким образом, ядро I не простаивало, а ядро II начало простаивать через 8 мс после старта и простаивало до конца вычислений через 12 мс после старта. Тогда суммарное время простоя всех ядер равно времени простоя ядра II и составляет  $12 - 8 = 4$  мс. Ответ: 4.

- 50) (А. Куканова) В файле **22-46.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Вычислительное устройство имеет **3 вычислительных ядра**. Каждый из параллельных процессов выполняется на отдельном ядре. Будем говорить, что ядро простаивает, если ядро не выполняет никакой процесс и при этом не вся совокупность процессов выполнена. Из готовых к выполнению процессов **в первую очередь запускаются процессы с наименьшим временем выполнения** (если таких больше, чем свободных ядер, приоритет имеют процессы с меньшими ID). Известно, что

выполнение всей совокупности процессов заняло минимальное возможное время. **Определите суммарное время простоя всех ядер.**

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |
| 5             | 5                                | 0                 |

Рассмотрим пример выше в случае, если устройство имеет 2 вычислительных ядра: ядро I и ядро II. Независимые процессы 1, 2 и 5 готовы к выполнению, но запустятся только процессы 1 (пусть на ядре I) и 2 (на ядре II) как имеющие меньшее время выполнения. При этом процесс 2 завершится через 3 мс. Поскольку процесс 3 может начаться только после завершения обоих процессов 1 и 2, на освободившемся ядре II запускается процесс 5, который завершится через  $3 + 5 = 8$  мс после старта. Процесс 1 завершится через 4 мс после старта и позволит начать выполнение процесса 3 на освободившемся ядре I. Процесс 3 завершится через  $4 + 1 = 5$  мс после старта. На освободившемся ядре I начнётся выполнение процесса 4, которое продлится 7 мс и закончится через  $5 + 7 = 12$  мс после начала вычислений. Таким образом, ядро I не простаивало, а ядро II начало простаивать через 8 мс после старта и простаивало до конца вычислений через 12 мс после старта. Тогда суммарное время простоя всех ядер равно времени простоя ядра II и составляет  $12 - 8 = 4$  мс. Ответ: 4.

- 51) (А. Куканова) В файле **22-46.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Вычислительное устройство имеет **4 вычислительных ядра**. Каждый из параллельных процессов выполняется на отдельном ядре. Готовые к выполнению **процессы добавляются в очередь**. Если в очередь одновременно добавляется несколько процессов, они располагаются в ней **в порядке возрастания ID**. Первый в очереди процесс запускается, как только появляется свободное ядро, и выходит из очереди (если остались свободные ядра, процесс повторяется). **Какой процесс завершился последним? В ответе укажите сумму его ID и времени, прошедшего с момента начала вычислений до их завершения.**

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |
| 5             | 5                                | 0                 |

Рассмотрим пример выше в случае, если устройство имеет 2 вычислительных ядра: ядро I и ядро II. Независимые процессы 1, 2 и 5 готовы к выполнению и располагаются в очереди в порядке возрастания ID. Запустятся процессы 1 (пусть на ядре I) и 2 (на ядре II), в очереди останется

процесс 5. При этом процесс 2 завершится через 3 мс, и освободившемся ядре II запускается единственный в очереди процесс 5, который завершится через  $3 + 5 = 8$  мс после старта. Очередь становится пуста. Процесс 1 завершится через 4 мс после старта и позволит добавить в очередь процесс 3, который сразу же начнёт выполнение на освободившемся ядре I. Очередь снова пуста. Процесс 3 завершится через  $4 + 1 = 5$  мс после старта. Процесс 4 встанет в очередь и сразу же начнёт выполняться на освободившемся ядре I. Выполнение процесса 4 продлится 7 мс и закончится через  $5 + 7 = 12$  мс после начала вычислений. Все процессы выполнены, последним завершился процесс 4 через 12 мс после старта. Ответом будет сумма 4 и 12, т.е. 16.

- 52) (А. Куканова) В файле **22-46.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Вычислительное устройство имеет **3 вычислительных ядра**. Каждый из параллельных процессов выполняется на отдельном ядре. Готовые к выполнению **процессы добавляются в очередь**. Если в очередь одновременно добавляется несколько процессов, они располагаются в ней **в порядке возрастания ID**. Первый в очереди процесс запускается, как только появляется свободное ядро, и выходит из очереди (если остались свободные ядра, процесс повторяется). **Какой процесс завершился последним? В ответе укажите сумму его ID и времени, прошедшего с момента начала вычислений до их завершения.**

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |
| 5             | 5                                | 0                 |

Рассмотрим пример выше в случае, если устройство имеет 2 вычислительных ядра: ядро I и ядро II. Независимые процессы 1, 2 и 5 готовы к выполнению и располагаются в очереди в порядке возрастания ID. Запустятся процессы 1 (пусть на ядре I) и 2 (на ядре II), в очереди останется процесс 5. При этом процесс 2 завершится через 3 мс, и освободившемся ядре II запускается единственный в очереди процесс 5, который завершится через  $3 + 5 = 8$  мс после старта. Очередь становится пуста. Процесс 1 завершится через 4 мс после старта и позволит добавить в очередь процесс 3, который сразу же начнёт выполнение на освободившемся ядре I. Очередь снова пуста. Процесс 3 завершится через  $4 + 1 = 5$  мс после старта. Процесс 4 встанет в очередь и сразу же начнёт выполняться на освободившемся ядре I. Выполнение процесса 4 продлится 7 мс и закончится через  $5 + 7 = 12$  мс после начала вычислений. Все процессы выполнены, последним завершился процесс 4 через 12 мс после старта. Ответом будет сумма 4 и 12, т.е. 16.

- 53) (А. Куканова) В файле **22-46.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в

миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Вычислительное устройство имеет **4 вычислительных ядра**. Каждый из параллельных процессов выполняется на отдельном ядре. Освободившееся ядро сразу же занимается готовым к выполнению процессом (если они есть), причём **в первую очередь запускаются процессы с наибольшим временем выполнения** (если таких больше, чем свободных ядер, приоритет имеют процессы с меньшими ID). **Определите наибольшее возможное число процессов, выполненных одним ядром. В ответе укажите сумму этого числа и времени, прошедшего с момента начала вычислений до их завершения.**

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |
| 5             | 5                                | 0                 |

Рассмотрим пример выше в случае, если устройство имеет 2 вычислительных ядра: ядро I и ядро II. Независимые процессы 1, 2 и 5 готовы к выполнению, но запустятся только процессы 5 (пусть на ядре I) и 1 (на ядре II) как имеющие большее время выполнения. При этом процесс 1 завершится через 4 мс после старта, и на освободившемся ядре II начнётся выполнение процесса 2. Оно продлится 3 мс и завершится через  $4 + 3 = 7$  мс после старта. Процесс 5 завершится через 5 мс после старта, но процесс 3 может быть запущен только после завершения процессов 1 и 2, поэтому ядро I останется свободным. Заметим, что процессы 3 и 4 выполняются последовательно и могут быть выполнены на одном и том же ядре. К моменту завершения процесса на ядре I был выполнен 1 процесс (ID 5), а на ядре II — 2 процесса (ID 1 и 2). Тогда максимальное число процессов, выполненных на одном ядре, будет достигнуто, если процессы 3 и 4 запустить на ядре II, и составит 4. Процесс 3 продлится 1 мс и закончит выполнение через  $7 + 1 = 8$  мс после старта. Процесс 4 продлится 7 мс и завершится через  $8 + 7 = 15$  мс после старта. Таким образом, вычисления завершились через 15 мс. Ответ будем сумма 15 и 4 (наибольшее число процессов на одном ядре), т.е. 19.

- 54) (А. Куканова) В файле **22-46.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0. Вычислительное устройство имеет **3 вычислительных ядра**. Каждый из параллельных процессов выполняется на отдельном ядре. Освободившееся ядро сразу же занимается готовым к выполнению процессом (если они есть), причём **в первую очередь запускаются процессы с наибольшим временем выполнения** (если таких больше, чем свободных ядер, приоритет имеют процессы с меньшими ID). **Определите наибольшее возможное число процессов, выполненных одним ядром. В ответе укажите сумму этого числа и времени, прошедшего с момента начала вычислений до их завершения.**

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |
| 5             | 5                                | 0                 |

Рассмотрим пример выше в случае, если устройство имеет 2 вычислительных ядра: ядро I и ядро II. Независимые процессы 1, 2 и 5 готовы к выполнению, но запустятся только процессы 5 (пусть на ядре I) и 1 (на ядре II) как имеющие большее время выполнения. При этом процесс 1 завершится через 4 мс после старта, и на освободившемся ядре II начнётся выполнение процесса 2. Оно продлится 3 мс и завершится через  $4 + 3 = 7$  мс после старта. Процесс 5 завершится через 5 мс после старта, но процесс 3 может быть запущен только после завершения процессов 1 и 2, поэтому ядро I останется свободным. Заметим, что процессы 3 и 4 выполняются последовательно и могут быть выполнены на одном и том же ядре. К моменту завершения процесса на ядре I был выполнен 1 процесс (ID 5), а на ядре II — 2 процесса (ID 1 и 2). Тогда максимальное число процессов, выполненных на одном ядре, будет достигнуто, если процессы 3 и 4 запустить на ядре II, и составит 4. Процесс 3 продлится 1 мс и закончит выполнение через  $7 + 1 = 8$  мс после старта. Процесс 4 продлится 7 мс и завершится через  $8 + 7 = 15$  мс после старта. Таким образом, вычисления завершились через 15 мс. Ответ будем сумма 15 и 4 (наибольшее число процессов на одном ядре), т.е. 19.

- 55) (А. Кабанов) В файле **22-55.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно. Для запуска некоторых процессов необходимы данные, которые получаются как результаты выполнения одного или двух других процессов – поставщиков данных. Независимые процессы (не имеющие поставщиков данных) можно запускать в любой момент времени. Если процесс В (зависимый процесс) получает данные от процесса А (поставщика данных), то процесс В может начать выполнение сразу же после завершения процесса А. Любые процессы, готовые к выполнению, можно запускать параллельно, при этом количество одновременно выполняемых процессов может быть любым, длительность процесса не зависит от других параллельно выполняемых процессов. В таблице представлены идентификатор (ID) каждого процесса, его длительность и ID поставщиков данных для зависимых процессов.

Определите, какое наибольшее количество процессов может быть завершено за первые 60 мс с момента запуска первого процесса.

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |

Пусть нас интересуют процессы, которые завершились в первые 6 мс. В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2, то есть, через 4 мс после старта. Он длится 1 мс и закончится через  $4 + 1 = 5$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса 3, то есть, через 5 мс. Он длится 7 мс и закончится через  $5 + 7 = 12$  мс. За первые 6 мс успеют завершиться процессы 1, 2, 3. Ответ для этого примера: 3.



56) (А. Кабанов) В файле **22-56.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Для запуска некоторых процессов необходимы данные, которые получаются как результаты выполнения одного или двух других процессов – поставщиков данных. Независимые процессы (не имеющие поставщиков данных) можно запускать в любой момент времени. Если процесс В (зависимый процесс) получает данные от процесса А (поставщика данных), то процесс В может начать выполнение сразу же после завершения процесса А. Любые процессы, готовые к выполнению, можно запускать параллельно, при этом количество одновременно выполняемых процессов может быть любым, длительность процесса не зависит от других параллельно выполняемых процессов. В таблице представлены идентификатор (ID) каждого процесса, его длительность и ID поставщиков данных для зависимых процессов.

Определите, какое наибольшее количество процессов может быть завершено за первые 40 мс с момента запуска первого процесса.

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |

Пусть нас интересуют процессы, которые завершились в первые 6 мс. В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2, то есть, через 4 мс после старта. Он длится 1 мс и закончится через  $4 + 1 = 5$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса 3, то есть, через 5 мс. Он длится 7 мс и закончится через  $5 + 7 = 12$  мс. За первые 6 мс успеют завершиться процессы 1, 2, 3. Ответ для этого примера: 3.

57) (Е. Джобс) В файле **22-57.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Для запуска некоторых процессов необходимы данные, которые получаются как результаты выполнения одного или двух других процессов – поставщиков данных. Независимые процессы (не имеющие поставщиков данных) можно запускать в любой момент времени. Если процесс В (зависимый процесс) получает данные от процесса А (поставщика данных), то процесс В может начать выполнение сразу же после завершения процесса А. Любые процессы, готовые к выполнению, можно запускать параллельно, при этом количество одновременно выполняемых процессов может быть любым, длительность процесса не зависит от других параллельно выполняемых процессов. В таблице представлены идентификатор (ID) каждого процесса, его длительность и ID поставщиков данных для зависимых процессов.

Определите максимально возможное целочисленное неизвестное время выполнения процесса  $t$ , если известно, что при его уменьшении минимально возможное время выполнения всей совокупности процессов не меняется.

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | $t$                              | 1; 2              |
| 4             | 7                                | 3                 |

Пусть выполнение данной совокупности процессов закончилось за 15 мс. В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2, то есть, через 4 мс после старта. Он длится  $t$  мс и закончится через  $4 + t$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса 3, то есть, через  $4 + t$  мс. Он длится 7 мс, так что минимальное время завершения всех процессов равно  $4 + t + 7 = 15$  мс. Следовательно,  $t = 15 - 4 - 7 = 4$  мс. Ответ для этого примера: 4.

- 58) (Е. Джобс) В файле **22-58.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Для запуска некоторых процессов необходимы данные, которые получаются как результаты выполнения одного или двух других процессов – поставщиков данных. Независимые процессы (не имеющие поставщиков данных) можно запускать в любой момент времени. Если процесс  $B$  (зависимый процесс) получает данные от процесса  $A$  (поставщика данных), то процесс  $B$  может начать выполнение сразу же после завершения процесса  $A$ . Любые процессы, готовые к выполнению, можно запускать параллельно, при этом количество одновременно выполняемых процессов может быть любым, длительность процесса не зависит от других параллельно выполняемых процессов. В таблице представлены идентификатор (ID) каждого процесса, его длительность и ID поставщиков данных для зависимых процессов.

Для одного из процесса неизвестно, после какого процесса  $x$  он должен начать работать. Известно, что минимальное время выполнения всех процессов равно 17 мс. Найдите номер процесса  $x$ .

Типовой пример организации данных в файле:

| ID процесса B | Время выполнения процесса B (мс) | ID процесса(ов) A |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 6                                | $x$               |
| 4             | 7                                | 3                 |

Пусть минимальное время выполнения данной совокупности процессов равно 10 мс. В данном случае минимальное время окончания процесса 1 – 4 мс от начала запуска процессов, процесса 2 – 3 мс, процесса 4 – 6 мс (3 мс + 3 мс), следовательно, третий процесс может завершиться за 10 мс. Так как время его выполнения 6 мс, то он должен начаться не позднее, чем через 4 мс после начал выполнения всех процессов. Через 4 мс заканчивается только один процесс – процесс 1.

Значит,  $x$  равен 1.

- 59) (А. Богданов) В файле **22-59.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- 60) (А. Богданов) В файле **22-60.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- Определите максимальное количество процессов, которые завершатся за 73 мс, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 61) (PRO100 ЕГЭ) В файле **22-61.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).
- Определите минимальное время, через которое может завершиться выполнение процесса с ID = 7, при условии, что все независимые друг от друга процессы могут выполняться параллельно.



- 62) В файле **22-62.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).

Определите, сколько процессов закончатся не позднее, чем через 200 мс после старта первого процесса при условии, что все независимые друг от друга процессы могут выполняться параллельно и каждый процесс начинает выполняться сразу, как только для этого есть возможность.

- 63) В файле **22-62.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).

Определите, сколько процессов будут активны на 200-й мс после старта первого процесса при условии, что все независимые друг от друга процессы могут выполняться параллельно и каждый процесс начинает выполняться сразу, как только для этого есть возможность.

- 64) В файле **22-62.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).

Определите, сколько процессов будут в состоянии ожидания на 150-й мс после старта первого процесса при условии, что все независимые друг от друга процессы могут выполняться параллельно и каждый процесс начинает выполняться сразу, как только для этого есть возможность.

- 65) (**А. Богданов**) Ваня пишет скрипты для генерации заданий ЕГЭ. Известно время, которое затрачивает Ваня на написание каждого скрипта, и время выполнения скрипта. Запуск очередного скрипта возможен только после окончания выполнения скриптов, которые подготавливают для него данные, т. е. тех, от которых скрипт зависит. Пока скрипт работает, Ваня может писать следующий скрипт, но Ваня не может писать два скрипта одновременно и пишет все скрипты в том порядке, в котором они внесены в таблицу. Компьютер у Вани многоядерный и может обрабатывать много простых скриптов одновременно, без взаимного влияния на общую производительность.

Информация о скриптах записана в файле **22-65.xls**. Типовой пример организации данных в файле:

| ID скрипта | Время написания скрипта (мин) | Время выполнения скрипта (мин) | ID скриптов-поставщиков данных |
|------------|-------------------------------|--------------------------------|--------------------------------|
| 1          | 5                             | 4                              | 0                              |
| 2          | 2                             | 3                              | 0                              |
| 3          | 7                             | 1                              | 1; 2                           |
| 4          | 4                             | 7                              | 3                              |

В данном случае скрипт 1 можно запустить только через 5 минут после начала работы, он закончит выполняться через  $5 + 4 = 9$  минут. Скрипт 2 Ваня напишет через  $5 + 2 = 7$  минут, он сразу начнёт работу и закончит выполняться через  $7 + 3 = 10$  минут. Таким образом, через 11 минут все скрипты-поставщики данных для процесса 3 уже закончили работу, но сам скрипт 3 Ваня напишет (и сможет запустить) только через  $5 + 2 + 7 = 14$  мин после начала работы. Этот скрипт закончит выполняться через  $14 + 1 = 15$  минут. Последний скрипт 4 Ваня напишет через  $14 + 4 = 18$  мин, он закончит выполняться через  $18 + 7 = 25$  мин. Ответ: 25.

- 66) В файле **22-62.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).

Определите количество процессов, выполнение которых начнётся не позднее, чем через 150 мс после запуска первого процесса.

67) В файле **22-62.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).

Определите количество процессов, выполнение которых начнётся не ранее, чем через 100 мс после запуска первого процесса.

68) В файле **22-62.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).

Определите количество процессов, которых будут активны на 100-й мс после запуска первого процесса.

69) **(А. Богданов)** В файле **22-69.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).

Определите количество процессов, которых будут активны на 32-й мс после запуска первого процесса.

70) **(А. Богданов)** В файле **22-69.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).

Определите количество процессов, которых завершатся не ранее чем через 32 мс после запуска первого процесса.

71) **(Е. Джобс)** В файле **22-71.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года).

Определите максимальное количество процессов, которых завершатся за 100 мс.

72) **(Е. Джобс)** В файле **22-72.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Для запуска некоторых процессов необходимы данные, которые получаются как результаты выполнения одного или двух других процессов – поставщиков данных. Независимые процессы (не имеющие поставщиков данных) можно запускать в любой момент времени. Если процесс  $B$  (зависимый процесс) получает данные от процесса  $A$  (поставщика данных), то процесс  $B$  может начать выполнение сразу же после завершения процесса  $A$ . Любые процессы, готовые к выполнению, можно запускать параллельно, при этом количество одновременно выполняемых процессов может быть любым, длительность процесса не зависит от других параллельно выполняемых процессов.

В таблице представлены идентификатор (ID) каждого процесса, его длительность и ID поставщиков данных для зависимых процессов.

Определите максимально возможное целочисленное неизвестное время выполнения процесса  $t$ , если известно, что вся совокупность процессов завершилась за 220 мс.

Типовой пример организации данных в файле:

| ID процесса B | Время выполнения процесса B (мс) | ID процесса(ов) A |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | $t$                              | 1; 2              |
| 4             | 7                                | 3                 |

Пусть выполнение данной совокупности процессов закончилось за 15 мс. В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2, то есть, через 4 мс после старта. Он длится  $t$  мс и закончится

через  $4 + t$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса 3, то есть, через  $4 + t$  мс. Он длится 7 мс, так что минимальное время завершения всех процессов равно  $4 + t + 7 = 15$  мс. Следовательно,  $t = 15 - 4 - 7 = 4$  мс. Ответ для этого примера: 4.

- 73) (Е. Джебс) В файле **22-73.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно. Для запуска некоторых процессов необходимы данные, которые получаются как результаты выполнения одного или двух других процессов – поставщиков данных. Независимые процессы (не имеющие поставщиков данных) можно запускать в любой момент времени. Если процесс В (зависимый процесс) получает данные от процесса А (поставщика данных), то процесс В может начать выполнение сразу же после завершения процесса А. Любые процессы, готовые к выполнению, можно запускать параллельно, при этом количество одновременно выполняемых процессов может быть любым, длительность процесса не зависит от других параллельно выполняемых процессов. В таблице представлены идентификатор (ID) каждого процесса, его длительность и ID поставщиков данных для зависимых процессов.

Известно, что при запуске описанной совокупности процессов произошла задержка, то есть все процессы начали выполняться не раньше, чем через  $X$  мс, при этом вся совокупность процессов завершилась через 300 мс после запуска. Определите максимально допустимое время задержки  $X$ .

Типовой пример организации данных в файле:

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 0                 |
| 3             | 1                                | 1; 2              |
| 4             | 7                                | 3                 |

Пусть эта совокупность процессов завершилась за 16 мс. Сначала предположим, что все процессы начались в момент 0. В данном случае независимые процессы 1 и 2 могут выполняться параллельно, при этом процесс 1 завершится через 4 мс, а процесс 2 – через 3 мс с момента старта. Процесс 3 может начаться только после завершения обоих процессов 1 и 2, то есть, через 4 мс после старта. Он длится 1 мс и закончится через  $4 + 1 = 5$  мс после старта. Выполнение процесса 4 может начаться только после завершения процесса 3, то есть, через 5 мс. Он длится 7 мс и закончится через  $5 + 7 = 12$  мс. Чтобы вся совокупность процессов завершилась за 16 секунд, задержка на старте не должна быть более  $16 - 12 = 4$  мс. Ответ для этого примера: 4.

- 74) (ЕГЭ-2023) В файле **22-74.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года). Определите минимальное время, через которое завершится выполнение всей совокупности процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 75) (ЕГЭ-2023) В файле **22-75.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года). Определите минимальное время, через которое завершится выполнение всей совокупности процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 76) (ЕГЭ-2023) В файле **22-76.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года). Определите минимальное время, через которое завершится выполнение всей совокупности процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.

- 77) (Е. Джобс) В файле **22-77.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года). Определите минимальное время, через которое завершится выполнение всей совокупности процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 78) (А. Рогов) В файле **22-78.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года). Определите минимальное время, через которое завершится выполнение всей совокупности процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 79) (А. Богданов) В файле **22-79.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2023 года). Определите минимальное время, через которое завершится выполнение всей совокупности процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 80) (К. Багдасарян) В файле **22-80.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Определите максимальную продолжительность отрезка времени (в мс), в течение которого возможно одновременное выполнение шести процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 81) (К. Багдасарян) В файле **22-81.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Вычислительные мощности оборудования не позволяют выполнять одновременно более 3 процессов. Определите минимальное время (в мс), через которое завершится выполнение всей совокупности процессов.
- 82) (К. Багдасарян) В файле **22-82.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Вычислительные мощности оборудования не позволяют выполнять одновременно более 4 процессов. Определите минимальное время (в мс), через которое завершится выполнение всей совокупности процессов.
- 83) (К. Багдасарян) В файле **22-83.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Вычислительные мощности оборудования не позволяют выполнять одновременно более 4 процессов. Определите минимальное время (в мс), через которое завершится выполнение всей совокупности процессов.
- 84) (PRO100-ЕГЭ) В файле **22-84.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Определите максимальную продолжительность отрезка времени (в мс), в течение которого возможно одновременное выполнение **четырёх** процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 85) (PRO100-ЕГЭ) В файле **22-85.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Определите максимальную продолжительность отрезка времени (в мс), в течение которого возможно одновременное выполнение **четырёх** процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.

- 86) (**PRO100-ЕГЭ**) В файле **22-86.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Определите максимальную продолжительность отрезка времени (в мс), в течение которого возможно одновременное выполнение **четырёх** процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 87) (**PRO100-ЕГЭ**) В файле **22-87.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Определите максимальную продолжительность отрезка времени (в мс), в течение которого возможно одновременное выполнение **четырёх** процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 88) (**PRO100-ЕГЭ**) В файле **22-88.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Определите максимальную продолжительность отрезка времени (в мс), в течение которого возможно одновременное выполнение **четырёх** процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 89) (**PRO100-ЕГЭ**) В файле **22-89.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Определите максимальную продолжительность отрезка времени (в мс), в течение которого возможно одновременное выполнение **пяти** процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 90) (**PRO100-ЕГЭ**) В файле **22-90.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Определите максимальную продолжительность отрезка времени (в мс), в течение которого возможно одновременное выполнение **четырёх** процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 91) (**PRO100-ЕГЭ**) В файле **22-91.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Определите максимальную продолжительность отрезка времени (в мс), в течение которого возможно одновременное выполнение **четырёх** процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 92) (**PRO100-ЕГЭ**) В файле **22-92.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Определите максимальную продолжительность отрезка времени (в мс), в течение которого возможно одновременное выполнение хотя бы **двух** процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 93) (**PRO100-ЕГЭ**) В файле **22-93.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Определите минимальное время выполнения всей совокупности процессов (в мс), если возможно параллельное выполнение максимум двух процессов. Выполнение процесса можно ставить на паузу и моментально переключаться на выполнение другого процесса.



- 94) **\*(PRO100-ЕГЭ)** В файле **22-94.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Определите максимальную продолжительность отрезка времени (в мс), в течение которого возможно одновременное выполнение хотя бы **трёх** процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 95) **(PRO100-ЕГЭ)** В файле **22-95.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Определите максимальную продолжительность отрезка времени (в мс), в течение которого возможно выполнение хотя бы **одного** процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 96) **(Г. Шапошников)** В файле **22-96.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года).

| ID процесса В | Время выполнения процесса В (мс) | ID процесса(ов) А |
|---------------|----------------------------------|-------------------|
| 1             | 4                                | 0                 |
| 2             | 3                                | 1                 |
| 3             | 1                                | 2                 |
| 4             | 7                                | 0                 |
| 5             | 6                                | 1; 4              |

Будем считать, что процессы выполняются одновременно, в случае если есть хотя бы один момент времени, когда оба процесса выполнялись. Найдите количество процессов, которые могли выполняться одновременно с процессом номер 13 (не обязательно все одновременно).

*Пояснение к примеру:* допустим нам нужно определить количество процессов, которые могли выполняться одновременно с процессом 2. Процессы 1 и 3 не подходят, т.к. процесс 2 можно запустить только после выполнения процесса 1, а процесс 3 только после выполнения процесса 2. Процесс 4 подходит в случае, если процесс 1 был запущен в самом начале, сразу после него был запущен процесс 2 (через 4 мс после начала работы программы) и процесс 4 также был запущен сразу после начала работы программы (тогда с 4 по 7 мс процессы 2 и 4 выполнялись совместно). Процесс 5 подходит, в случае, если запустить процесс 4 в самом начале, и сразу после него (через 7 мс после начала работы программы) запустить процесс 5. Процесс 1 же можно запустить через 3 мс после начала работы программы (тогда закончится он к 7 мс). И сразу после него запустить процесс 2. Тогда с 7 мс по 10 мс процесс 2 и 5 будут выполняться совместно.

Таким образом, одновременно с процессом 2 могут выполняться процессы 4 и 5. Ответ: 2.

- 97) **(Л. Шастин)** В файле **22-97.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Определите максимальную продолжительность отрезка времени (в мс), в течение которого возможно одновременное выполнение **четырёх** процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.
- 98) **(Л. Шастин)** В файле **22-98.xls** содержится информация о совокупности N вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Определите максимальную продолжительность отрезка времени (в мс), в течение которого возможно одновременное выполнение **пяти** процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.

- 99) (**Л. Шастин**) В файле **22-99.xls** содержится информация о совокупности  $N$  вычислительных процессов, которые могут выполняться параллельно или последовательно... (Условие совпадает с условием задачи из демо-варианта 2024 года). Определите максимальную продолжительность отрезка времени (в мс), в течение которого возможно одновременное выполнение **шести** процессов, при условии, что все независимые друг от друга процессы могут выполняться параллельно.

## 22 (повышенный уровень, время – 7 мин)

**Тема:** Анализ программы, содержащей подпрограммы, циклы и ветвления.

**Что проверяется:**

Умение анализировать алгоритм, содержащий ветвление и цикл

1.6.1. Формализация понятия алгоритма.

1.1.4. Умение читать и отлаживать программы на языке программирования.

**Что нужно знать:**

- операции целочисленного деления (/) и взятия остатка (%)
- как работают операторы присваивания, циклы и условные операторы в языке программирования

### Задачи для тренировки<sup>1</sup>:

- 1) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
#include <iostream>
using namespace std;
int main()
{
 int L = 0, M = 0, x;
 cin >> x;
 while (x > 0) {
 L = L + 1;
 M = M + (x % 10);
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}
```

- 2) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```
#include <iostream>
using namespace std;
int main()
{
 int L = 0, M = 0, x;
 cin >> x;
 while (x > 0) {
 L = L + 1;
 if (x % 2 == 0)
 M = M + (x % 10);
 x = x / 10;
 }
```

<sup>1</sup> Источники заданий:

1. Авторские разработки.
2. Тренировочные и диагностические работы МИОО.
3. Путимцева Ю.С. Информатика. Диагностические работы в формате ЕГЭ 2012. М.: МЦНМО, 2012.
4. Евич Л.Н., Кулабухов С.Ю. Информатика и ИКТ. Подготовка к ЕГЭ-2014. — Ростов-на-Дону: Легион, 2013.



```

 }
 cout << L << endl << M;
 return 0;
}

```

- 3) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 0.

```

#include <iostream>
using namespace std;
int main()
{
 int L = 0, M = 0, x;
 cin >> x;
 while (x > 0) {
 L = L + 1;
 if (x % 2 == 0)
 M = M + (x % 10);
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}

```

- 4) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```

#include <iostream>
using namespace std;
int main()
{
 int L = 0, M = 0, x; cin >> x;
 while (x > 0) {
 L = L + 1;
 if (x % 2 == 1)
 M = M + (x % 10);
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}

```

- 5) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```

#include <iostream>
using namespace std;
int main()
{
 int L = 0, M = 0, x;
 cin >> x;
 while (x > 0) {
 L = L + 1;
 if (x % 2 == 0)

```

```

 M = M + (x % 10) / 2;
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}

```

- 6) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```

#include <iostream>
using namespace std;
int main()
{
 int L = 0, M = 0, x;
 cin >> x;
 while (x > 0) {
 L = L + 1;
 if (x % 2 == 1)
 M = M + (x % 10) / 2;
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}

```

- 7) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```

#include <iostream>
using namespace std;
int main()
{
 int L = 0, M = 0, x;
 cin >> x;
 while (x > 0) {
 L = L+1;
 if (M < x)
 M = (x % 10);
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}

```

- 8) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```

#include <iostream>
using namespace std;
int main()
{
 int L = 0, M = 0, x;
 cin >> x;

```

---

```

while (x > 0) {
 L = L+1;
 if ((M < x) && (x % 2 == 0))
 M = x % 10;
 x = x / 10;
}
cout << L << endl << M;
return 0;
}

```

- 9) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 10.

```

#include <iostream>
using namespace std;
int main()
{
 int L = 0, M = 0, x;
 cin >> x;
 while (x > 0) {
 L = L+1;
 if ((M < x) && (x % 2 == 1))
 M = (x % 10) * 2;
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}

```

- 10) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 10.

```

#include <iostream>
using namespace std;
int main()
{
 int L = 0, M = 0, x;
 cin >> x;
 while (x > 0) {
 L = L+1;
 if (M < x)
 M = (x % 10) * 2;
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}

```

- 11) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 72.

```

#include <iostream>
using namespace std;
int main()

```

---

```

{
 int a = 0, b = 1, x;
 cin >> x;
 while (x>0) {
 a = a+1;
 b = b*(x % 10);
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}

```

- 12) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 14.

```

#include <iostream>
using namespace std;
int main()
{
 int a = 0, b = 1, x;
 cin >> x;
 while (x > 0) {
 a = a + 1;
 b = b * (x % 10);
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}

```

- 13) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```

#include <iostream>
using namespace std;
int main()
{
 int a = 0, b = 1, x;
 cin >> x;
 while (x > 0) {
 a = a + 1;
 b = b * (x % 10);
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}

```

- 14) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 0.

```

#include <iostream>
using namespace std;
int main()

```

---

```

{
 int a = 0, b = 1, x;
 cin >> x;
 while (x > 0) {
 a = a + 1;
 b = b * (x % 10);
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}

```

- 15) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```

#include <iostream>
using namespace std;
int main()
{
 int L = 0, M = 0, x;
 cin >> x;
 while (x > 0) {
 L = L + 1;
 M = M + (x % 10);
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}

```

- 16) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```

#include <iostream>
using namespace std;
int main()
{
 int L = 0, M = 0, x;
 cin >> x;
 while (x > 0) {
 L = L + 1;
 if (x % 2 == 0)
 M = M + (x % 10);
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}

```

- 17) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 0.

```

#include <iostream>
using namespace std;

```

```
int main()
{
 int L = 0, M = 0, x;
 cin >> x;
 while (x > 0) {
 L = L + 1;
 if (x % 2 == 0)
 M = M + (x % 10);
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}
```

- 18) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```
#include <iostream>
using namespace std;
int main()
{
 int L, M, x;
 cin >> x;
 L = 0; M = 0;
 while (x > 0) {
 L = L + 1;
 if (x % 2 == 1)
 M = M + (x % 10);
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}
```

- 19) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
#include <iostream>
using namespace std;
int main()
{
 int L, M, x;
 cin >> x;
 L = 0; M = 0;
 while (x > 0) {
 L = L + 1;
 if (x % 2 == 0)
 M = M + (x % 10) / 2;
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}
```

- 20) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
#include <iostream>
using namespace std;
int main()
{
 int L, M, x;
 cin >> x;
 L = 0; M = 0;
 while (x > 0) {
 L = L + 1;
 if (x % 2 == 1)
 M = M + (x % 10) / 2;
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}
```

- 21) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
#include <iostream>
using namespace std;
int main()
{
 int L, M, x;
 cin >> x;
 L = 0; M = 0;
 while (x > 0) {
 L = L+1;
 if (M < x)
 M = (x % 10);
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}
```

- 22) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```
#include <iostream>
using namespace std;
int main()
{
 int L, M, x;
 cin >> x;
 L = 0; M = 0;
 while (x > 0) {
 L = L+1;
 if ((M < x) && (x % 2 == 0))
```



```

 M = (x % 10);
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}

```

- 23) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 10.

```

#include <iostream>
using namespace std;
int main()
{
 int L, M, x;
 cin >> x;
 L = 0; M = 0;
 while (x > 0) {
 L = L+1;
 if ((M < x) && (x % 2 == 1))
 M = (x % 10) * 2;
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}

```

- 24) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 28.

```

#include <iostream>
using namespace std;
int main()
{
 int L, M, x;
 cin >> x;
 L = 0; M = 0;
 while (x > 0) {
 L = L+1;
 if (M < x)
 M = M + (x % 10) * 2;
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}

```

- 25) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 72.

```

#include <iostream>
using namespace std;
int main()
{

```

```

int a, b, x;
cin >> x;
a = 0; b = 1;
while (x>0) {
 a = a+1;
 b = b*(x % 10);
 x = x / 10;
}
cout << a << endl << b;
return 0;
}

```

- 26) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 14.

```

#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 a = a + 1;
 b = b * (x % 10);
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}

```

- 27) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```

#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 a = a + 1;
 b = b * (x % 10);
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}

```

- 28) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 36.

```

#include <iostream>

```

```
using namespace std;
int main()
{
 int a, b, x;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 a = a + 1;
 b = b * (x % 10);
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}
```

- 29) (Д.Ю. Мельникова, г. Саратов) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 24.

```
#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 a = a + 1;
 b = b * (x % 8);
 x = x / 8;
 }
 cout << a << endl << b;
 return 0;
}
```

- 30) (Д.Ю. Мельникова, г. Саратов) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 10.

```
#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 a = a + 1;
 b = b * (x % 8);
 x = x / 8;
 }
 cout << a << endl << b;
 return 0;
}
```

- 31) (Д.Ю. Мельникова, г. Саратов) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 6.

```
#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 a = a + 1;
 b = b * (x % 6);
 x = x / 6;
 }
 cout << a << endl << b;
 return 0;
}
```

- 32) (Д.Ю. Мельникова, г. Саратов) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 9.

```
#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 a = a + 1;
 b = b * (x % 5);
 x = x / 5;
 }
 cout << a << endl << b;
 return 0;
}
```

- 33) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 21.

```
#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin >> x;
 a = 0; b = 1;
 while (x>0) {
 a = a+1;
 b = b*(x % 10);
 x = x / 10;
 }
}
```

```

 cout << a << endl << b;
 return 0;
}

```

- 34) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 35.

```

#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin >> x;
 a = 0; b = 1;
 while (x>0) {
 a = a+1;
 b = b*(x % 10);
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}

```

- 35) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 4.

```

#include <iostream>
using namespace std;
int main()
{
 int L, M, x;
 cin >> x;
 L = 0; M = 9;
 while (x > 5) {
 L = L + 1;
 if (M > (x % 10)) M = (x % 10);
 x = x / 10;
 }
 cout << L << endl << M;
 return 0;
}

```

- 36) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```

#include <iostream>
using namespace std;
int main()
{
 int L, M, x;
 cin >> x;
 L = 0; M = 0;
 while (x>5) {
 L = L + 1;

```

---

```

 if (M < (x % 10)) M = (x % 10);
 x = x / 10;
}
cout << L << endl << M;
return 0;
}

```

- 37) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 6, а потом 5.

```

#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin >> x;
 a = 0; b = 0;
 while (x>0) {
 a = a + 2;
 b = b + (x % 10);
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}

```

- 38) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 8, а потом 19.

```

#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin >> x;
 a = 0; b = 0;
 while (x > 0) {
 a = a + 2;
 b = b + (x % 10);
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}

```

- 39) Ниже записан алгоритм. После выполнения алгоритма было напечатано 3 числа. Первые два напечатанных числа – это числа 7 и 42. Какое наибольшее число может быть напечатано третьим?

```

#include <iostream>
using namespace std;
int main()
{
 int r, a, b, x, y, z;
 cin >> x >> y;

```

---

```

 if (y > x) {
 z = x; x = y; y = z;
 }
 a = x; b = y;
 while (b > 0)
 {
 r = a % b;
 a = b;
 b = r;
 }
 cout << a << endl << x << endl << y;
 return 0;
}

```

- 40) Ниже записан алгоритм. После выполнения алгоритма было напечатано 3 числа. Первые два напечатанных числа – это числа 13 и 65. Какое наибольшее число может быть напечатано третьим?

```

#include <iostream>
using namespace std;
int main()
{
 int r, a, b, x, y, z;
 cin >> x >> y;
 if (y > x) {
 z = x; x = y; y = z;
 }
 a = x; b = y;
 while (b > 0) {
 r = a % b;
 a = b;
 b = r;
 }
 cout << a << endl << x << endl << y;
 return 0;
}

```

- 41) Ниже записан алгоритм. Укажите наименьшее из таких чисел N, при вводе которых алгоритм напечатает 17.

```

#include <iostream>
using namespace std;
int main()
{
 int N;
 cin >> N;
 for (int i = 1; i < N; i++)
 if (N % i == 0)
 q = i;
 cout << q;
 return 0;
}

```



- 42) (<http://ege.yandex.ru>) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите, сколько есть таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 12.

```
#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 a = a + 1;
 b = b * (x % 10);
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}
```

- 43) (<http://ege.yandex.ru>) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 8.

```
#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin >> x;
 a = 0; b = 0;
 while (x > 0) {
 a = a + 1;
 b = b + (x % 100);
 x = x / 100;
 }
 cout << a << endl << b;
 return 0;
}
```

- 44) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 8.

```
#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 a = a + 1;
 b = b * (x % 100);
 x = x / 100;
 }
}
```

```
 cout << a << endl << b;
 return 0;
}
```

- 45) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 14, а потом 6.

```
#include <iostream>
using namespace std;
int main()
{
 int a, b, x, c;
 cin >> x;
 a = 0; b = 10;
 while (x>0) {
 c = (x % 10);
 a = a + c;
 if (c < b) b = c;
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}
```

- 46) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 15, а потом 5.

```
#include <iostream>
using namespace std;
int main()
{
 int a, b, c, x;
 cin >> x;
 a = 0; b = 10;
 while (x>0) {
 c = (x % 10);
 a = a + c;
 if (c < b) b = c;
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}
```

- 47) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 13, а потом 3.

```
#include <iostream>
using namespace std;
int main()
{
 int a, b, c, x;
 cin >> x;
 a = 0; b = 10;
```

```

while (x>0) {
 c = (x % 10);
 a = a + c;
 if (c < b) b = c;
 x = x / 10;
}
cout << a << endl << b;
return 0;
}

```

- 48) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 19, а потом 4.

```

#include <iostream>
using namespace std;
int main()
{
 int a, b, c, x;
 cin >> x;
 a = 0; b = 10;
 while (x>0) {
 c = (x % 10);
 a = a + c;
 if (c < b) b = c;
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}

```

- 49) Ниже записан алгоритм. Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 2.

```

#include <iostream>
using namespace std;
int main()
{
 int a, b, c, x;
 cin >> x;
 a = 0; b = 0;
 while (x > 0) {
 c = x % 2;
 if (c == 0) a = a + 1;
 else b = b + 1;
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}

```

- 50) Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 0.

```

#include <iostream>
using namespace std;
int main()

```

---

```

{
 int a, b, c, x;
 cin >> x;
 a = 0; b = 0;
 while (x > 0) {
 c = x % 2;
 if (c == 0) a = a + 1;
 else b = b + 1;
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}

```

- 51) Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 2.

```

#include <iostream>
using namespace std;
int main()
{
 int a, b, c, x;
 cin >> x;
 a = 0; b = 0;
 while (x > 0) {
 c = x % 2;
 if (c == 0)
 a = a + 1;
 else b = b + 1;
 x = x / 8;
 }
 cout << a << endl << b;
 return 0;
}

```

- 52) Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 0.

```

#include <iostream>
using namespace std;
int main()
{
 int a, b, c, x;
 cin >> x;
 a = 0; b = 0;
 while (x > 0) {
 c = x % 2;
 if (c == 0) a = a + 1;
 else b = b + 1;
 x = x / 6;
 }
 cout << a << endl << b;
 return 0;
}

```

- 53) Получив на вход число  $x$ , этот алгоритм печатает два числа  $K$  и  $R$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 3.

```
#include <iostream>
using namespace std;
int main()
{
 int x, K, R, y;
 cin >> x;
 K = 0; R = 9;
 y = (x % 10);
 while (x > 0) {
 K = K + 1;
 if (R > (x % 10)) R = (x % 10);
 x = x / 10;
 }
 R = y - R;
 cout << K << endl << R;
 return 0;
}
```

- 54) Получив на вход число  $x$ , этот алгоритм печатает два числа  $K$  и  $R$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
#include <iostream>
using namespace std;
int main()
{
 int x, K, R, y;
 cin >> x;
 K = 0; R = 9;
 y = (x % 10);
 while (x > 0) {
 K = K + 1;
 if (R > (x % 10)) R = (x % 10);
 x = x / 10;
 }
 R = y - R;
 cout << K << endl << R;
 return 0;
}
```

- 55) Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 13.

```
#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin << x;
 a = 0; b = 0;
 while (x > 0) {
 a = a+1;
 b = b+(x % 100);
 x = x / 100;
 }
```

```
 }
 cout << a << endl << b;
 return 0;
}
```

- 56) Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 15.

```
#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin << x;
 a = 0; b = 0;
 while (x > 0) {
 a = a+1;
 b = b+(x % 100);
 x = x / 100;
 }
 cout << a << endl << b;
 return 0;
}
```

- 57) Ниже записан алгоритм. Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 15?

```
#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin >> x;
 a = 0; b = 0;
 while (x>0) {
 a = a + 1;
 b = b + (x % 10);
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}
```

- 58) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает числа:  $a$  и  $b$ . Укажите наименьшее положительное пятизначное число  $x$ , при котором после выполнения алгоритма будет напечатано сначала 5, а потом 2.

```
#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 a = 0;
 b = 10;
 cin >> x;
 while (x > 0) {
 y = (x % 10);
```

```

 x = x / 10;
 if (y > a) a = y;
 if (y < b) b = y;
 }
 cout << a << endl << b;
 return 0;
}

```

59) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает числа:  $a$  и  $b$ .

Укажите наименьшее из таких чисел  $x$ , при вводе которого после выполнения алгоритма будет напечатано сначала 2, а потом 22.

```

#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin >> x;
 a = 0; b = 0;
 while (x > 0) {
 a = a+1;
 b = b+(x % 100);
 x = x / 100;
 }
 cout << a << endl << b;
 return 0;
}

```

60) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ .

Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 5.

```

#include <iostream>
using namespace std;
int main()
{
 int a, b, x;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 a = a+1;
 b = b*(x % 100);
 x = x / 100;
 }
 cout << a << endl << b;
 return 0;
}

```

61) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ .

Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 18.

```

#include <iostream>
using namespace std;
int main()
{
 int a, b, x;

```



---

```

cin >> x;
a = 0; b = 1;
while (x > 0) {
 a = a+1;
 b = b*(x % 100);
 x = x / 100;
}
cout << a << endl << b;
return 0;
}

```

62) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ .

Укажите наименьшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 5, а потом 3.

```

#include <iostream>
using namespace std;
int main()
{
 long int x, y, a, b;
 a = 0;
 b = 0;
 cin >> x;
 while (x > 0) {
 y = x % 10;
 if (y > 3)
 a = a + 1;
 if (y < 8)
 b = b + 1;
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}

```

63) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ .

Укажите наибольшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 4, а потом 3.

```

#include <iostream>
using namespace std;
int main()
{
 long int x, y, a, b;
 a = 0;
 b = 0;
 cin >> x;
 while (x > 0) {
 y = x % 10;
 if (y > 3)
 a = a + 1;
 if (y < 8)
 b = b + 1;
 x = x / 10;
 }
}

```

```
 cout << a << endl << b;
 return 0;
}
```

64) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ .

Укажите наибольшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 5, а потом 4.

```
#include <iostream>
using namespace std;
int main()
{
 long int x, y, a, b;
 a = 0;
 b = 0;
 cin >> x;
 while (x > 0) {
 y = x % 10;
 if (y > 4)
 a = a + 1;
 if (y < 6)
 b = b + 1;
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}
```

65) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ .

Укажите наименьшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 3, а потом 4.

```
#include <iostream>
using namespace std;
int main()
{
 long int x, y, a, b;
 a = 0;
 b = 0;
 cin >> x;
 while (x>0) {
 y = x % 10;
 if (y > 4)
 a = a + 1;
 if (y < 6)
 b = b + 1;
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}
```

66) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ .

Укажите наименьшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 2, а потом 4.

```
#include <iostream>
```

```
using namespace std;
int main()
{
 long int x, y, a, b;
 a = 0;
 b = 0;
 cin >> x;
 while (x>0) {
 y = x % 10;
 if (y > 3)
 a = a + 1;
 if (y < 7)
 b = b + 1;
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}
```

67) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ .

Укажите наибольшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 2, а потом 4.

```
#include <iostream>
using namespace std;
int main()
{
 long int x, y, a, b;
 a = 0;
 b = 0;
 cin >> x;
 while (x>0) {
 y = x % 10;
 if (y > 4)
 a = a + 1;
 if (y < 7)
 b = b + 1;
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}
```

68) Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 6.

```
#include <iostream>
using namespace std;
int main()
{
 int x, L, M;
 cin >> x;
 L = 0;
 M = 0;
 while (x > 0) {
```

```

 M = M + 1;
 if (x % 2 != 0)
 L = L + 1;
 x = x / 2;
 }
 cout << L << endl << M;
 return 0;
}

```

- 69) Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 11.

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0;
 b = 1;
 while (x > 0) {
 a = a + 1;
 b = b * (x % 1000);
 x = x / 1000;
 }
 cout << a << endl << b;
 return 0;
}

```

- 70) Ниже записан алгоритм. Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 13.

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0;
 b = 1;
 while (x > 0) {
 a = a + 2;
 b = b * (x % 1000);
 x = x / 1000;
 }
 cout << a << endl << b;
 return 0;
}

```

- 71) Ниже записан алгоритм. Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 12?

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b;

```

```
cin >> x;
a = 0;
b = 0;
while (x > 0) {
 a = a + 1;
 b = b + (x % 100);
 x = x / 100;
}
cout << a << endl << b;
return 0;
}
```

- 72) Получив на вход число **x**, этот алгоритм печатает два числа: **L** и **M**. Укажите наибольшее число **x**, при вводе которого алгоритм печатает сначала 5, а потом 8.

```
#include <iostream>
using namespace std;
int main()
{
 int x, L, M;
 cin >> x;
 L = 0;
 M = 0;
 while (x > 0) {
 M = M + 1;
 if (x % 2 != 0)
 L = L + 1;
 x = x / 2;
 }
 cout << L << endl << M;
 return 0;
}
```

- 73) Получив на вход число **x**, этот алгоритм печатает число **M**. Известно, что **x** > 100. Укажите наименьшее такое (т. е. большее 100) число **x**, при вводе которого алгоритм печатает 15.

```
#include <iostream>
using namespace std;
int main()
{
 int x, L, M;
 cin >> x;
 L = x - 30;
 M = x + 30;
 while (L != M) {
 if (L > M)
 L = L - M;
 else
 M = M - L;
 }
 cout << M;
 return 0;
}
```

- 74) Получив на вход число **x**, этот алгоритм печатает число **M**. Известно, что **x** > 100. Укажите наименьшее такое (т. е. большее 100) число **x**, при вводе которого алгоритм печатает 30.

```
#include <iostream>
using namespace std;
int main()
{
 int x, L, M;
 cin >> x;
 L = x - 30;
 M = x + 30;
 while (L != M) {
 if (L > M)
 L = L - M;
 else
 M = M - L;
 }
 cout << M;
 return 0;
}
```

- 75) Получив на вход число **x**, этот алгоритм печатает два числа **a** и **b**. Укажите наименьшее из таких чисел **x**, при вводе которых алгоритм печатает сначала 3, а потом 18.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0;
 b = 0;
 while (x > 0) {
 a = a + 1;
 if (x % 2 == 0)
 b = b + x % 10;
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}
```

- 76) Получив на вход число **x**, этот алгоритм печатает два числа **a** и **b**. Укажите наименьшее из таких чисел **x**, при вводе которых алгоритм печатает сначала 45, а потом 5.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b, c;
 cin >> x;
 a = 1;
 b = 10;
 while (x > 0) {
 c = x % 10;
 a = a * c;
 if (c < b)
 b = c;
 }
}
```

---

```

 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}

```

- 77) Получив на вход число **x**, этот алгоритм печатает число **M**. Известно, что **x** > 100. Укажите наименьшее такое (т. е. большее 100) число **x**, при вводе которого алгоритм печатает 2.

```

#include <iostream>
using namespace std;
int main()
{
 int x, L, M;
 cin >> x;
 L = x - 12;
 M = x + 12;
 while (L != M) {
 if (L > M)
 L = L - M;
 else
 M = M - L;
 }
 cout << M;
 return 0;
}

```

- 78) Получив на вход число **x**, этот алгоритм печатает число **M**. Известно, что **x** > 100. Укажите наименьшее такое (т. е. большее 100) число **x**, при вводе которого алгоритм печатает 11.

```

#include <iostream>
using namespace std;
int main()
{
 int x, L, M;
 cin >> x;
 L = x - 21;
 M = x + 12;
 while (L != M) {
 if (L > M)
 L = L - M;
 else
 M = M - L;
 }
 cout << M;
 return 0;
}

```

- 79) Получив на вход число **x**, этот алгоритм печатает число **M**. Известно, что **x** > 100. Укажите наименьшее такое (т. е. большее 100) число **x**, при вводе которого алгоритм печатает 35.

```

#include <iostream>
using namespace std;
int main()
{
 int x, L, M;
 cin >> x;

```



```
L = x - 15;
M = x + 20;
while (L != M)
{
 if (L > M)
 L = L - M;
 else
 M = M - L;
}
cout << M;
return 0;
}
```

- 80) Получив на вход число **x**, этот алгоритм печатает число **M**. Известно, что **x** > 100. Укажите наименьшее такое (т. е. большее 100) число **x**, при вводе которого алгоритм печатает 9.

```
#include <iostream>
using namespace std;
int main()
{
 int x, L, M;
 cin >> x;
 L = x - 18;
 M = x + 36;
 while (L != M) {
 if (L > M)
 L = L - M;
 else
 M = M - L;
 }
 cout << M;
 return 0;
}
```

- 81) Получив на вход число **x**, этот алгоритм печатает число **M**. Известно, что **x** > 100. Укажите наименьшее такое (т. е. большее 100) число **x**, при вводе которого алгоритм печатает 35.

```
#include <iostream>
using namespace std;
int main()
{
 int x, L, M;
 cin >> x;
 L = x - 20;
 M = x + 15;
 while (L != M) {
 if (L > M)
 L = L - M;
 else
 M = M - L;
 }
 cout << M;
 return 0;
}
```

- 82) Получив на вход число  $x$ , этот алгоритм печатает число  $M$ . Известно, что  $x > 100$ . Укажите наименьшее такое (т. е. большее 100) число  $x$ , при вводе которого алгоритм печатает 4.

```
#include <iostream>
using namespace std;
int main()
{
 int x, L, M;
 cin >> x;
 L = x - 16;
 M = x + 32;
 while (L != M) {
 if (L > M)
 L = L - M;
 else
 M = M - L;
 }
 cout << M;
 return 0;
}
```

- 83) Получив на вход число  $x$ , этот алгоритм печатает число  $M$ . Известно, что  $x > 100$ . Укажите наименьшее такое (т. е. большее 100) число  $x$ , при вводе которого алгоритм печатает 16.

```
#include <iostream>
using namespace std;
int main()
{
 int x, L, M;
 cin >> x;
 L = x - 16;
 M = x + 16;
 while (L != M) {
 if (L > M)
 L = L - M;
 else
 M = M - L;
 }
 cout << M;
 return 0;
}
```

- 84) (Д.Ф. Муфаззалов) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 2.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a = 0, b = 0, i = 0, c;
 cin >> x;
 while (x > 0) {
 i = i + 1;
 c = x % 10;
 if (i % 2 == 0)
 a = a + c;
 else
```

```

 b = b + c;
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}

```

- 85) (Д.Ф. Муфаззалов) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 5.

```

#include <iostream>
using namespace std;
int main()
{
 int x, b = 0, i = 0;
 cin >> x;
 while (x > 0) {
 if (i % 2 == 0)
 b = b + x % 10;
 x = x / 10;
 i = i + 1;
 }
 cout << i << endl << b;
 return 0;
}

```

- 86) (Д.Ф. Муфаззалов) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 5.

```

#include <iostream>
using namespace std;
int main()
{
 int x, b = 0, i = 0;
 cin >> x;
 while (x > 0) {
 b = b + x % 10;
 x = x / 10;
 i = i + 1;
 }
 cout << i << endl << b;
 return 0;
}

```

- 87) (Д.Ф. Муфаззалов) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 2.

```

#include <iostream>
using namespace std;
int main()
{
 int x, a = 0, b = 0, i = 0, c = 0;
 cin >> x;
 while (x > 0) {
 i = i + 1;
 if (i % 2 == 0)
 a = a + c;
 else

```

```
 b = b + c;
 c = x % 10;
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}
```

- 88) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 15.

```
#include <iostream>
using namespace std;
int main()
{ int x, a = 0, b = 0, d = 0;
 cin >> x;
 while (x > 0) {
 if (d % 2 == 0)
 a = a + x % 10;
 else
 b = b + x % 10;
 x = x / 10;
 d = d + 1;
 }
 cout << a << endl << b;
 return 0;
}
```

- 89) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 14, а потом 12.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a = 0, b = 0, d = 0;
 cin >> x;
 while (x > 0) {
 if (d % 2 == 0)
 a = a + x % 10;
 else
 b = b + x % 10;
 x = x / 10;
 d = d + 1;
 }
 cout << a << endl << b;
 return 0;
}
```

- 90) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , большее, чем 200, при вводе которого алгоритм напечатает 70.

```
#include <iostream>
using namespace std;
int main()
{
```

---

```

int x, L, M;
cin >> x;
L = 2*x - 30;
M = 2*x + 40;
while (L != M)
 if (L > M)
 L = L - M;
 else
 M = M - L;
cout << M;
return 0;
}

```

- 91) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , большее, чем 200, при вводе которого алгоритм напечатает 50.

```

#include <iostream>
using namespace std;
int main()
{
 int x, L, M;
 cin >> x;
 L = 2*x - 20;
 M = 2*x + 30;
 while (L != M)
 if (L > M)
 L = L - M;
 else
 M = M - L;
 cout << M;
 return 0;
}

```

- 92) Ниже приведён алгоритм. Укажите наименьшее натуральное число  $x$ , при вводе которого алгоритм напечатает четырёхзначное число.

```

#include <iostream>
using namespace std;
int main()
{
 int x, d, x0, N = 0;
 cin >> x;
 x0 = x;
 while (x > 0) {
 d = x%2;
 N = 10*N + d;
 x = x/2;
 }
 N = N + x0;
 cout << N;
 return 0;
}

```

- 93) Ниже приведён алгоритм. Укажите наименьшее натуральное число  $x$ , при вводе которого алгоритм напечатает пятизначное число.

```

#include <iostream>

```

```
using namespace std;
int main()
{
 int x, d, x0, N = 0;
 cin >> x;
 x0 = x;
 while (x > 0) {
 d = x%2;
 N = 10*N + d;
 x = x/2;
 }
 N = N + x0;
 cout << N;
 return 0;
}
```

- 94) Ниже приведён алгоритм. Укажите наименьшее натуральное число  $x$ , при вводе которого алгоритм напечатает четырёхзначное число.

```
#include <iostream>
using namespace std;
int main()
{
 int x, d, x0, N = 0;
 cin >> x;
 x0 = x;
 while (x > 0) {
 d = x%3;
 N = 10*N + d;
 x = x/3;
 }
 N = N + x0;
 cout << N;
 return 0;
}
```

- 95) Ниже приведён алгоритм. Укажите наименьшее натуральное число  $x$ , при вводе которого алгоритм напечатает пятизначное число.

```
#include <iostream>
using namespace std;
int main()
{
 int x, d, x0, N = 0;
 cin >> x;
 x0 = x;
 while (x > 0) {
 d = x%3;
 N = 10*N + d;
 x = x/3;
 }
 N = N + x0;
 cout << N;
 return 0;
}
```

- 96) Ниже приведён алгоритм. Укажите наименьшее натуральное число  $x$ , при вводе которого алгоритм напечатает шестизначное число.

```
#include <iostream>
using namespace std;
int main()
{
 int x, d, x0, N = 0;
 cin >> x;
 x0 = x;
 while (x > 0) {
 d = x%3;
 N = 10*N + d;
 x = x/3;
 }
 N = N + x0;
 cout << N;
 return 0;
}
```

- 97) (А.Н. Носкин) Ниже приведён алгоритм. Укажите наименьшее число  $x$ , при вводе которого алгоритм напечатает число, большее 10200.

```
#include <iostream>
using namespace std;
int main()
{
 int x, d, x0, N = 0;
 cin >> x;
 x0 = x;
 while (x > 0) {
 d = x%5;
 N = 10*N + d;
 x = x/5;
 }
 N = N + x0;
 cout << N;
 return 0;
}
```

- 98) (А.Н. Носкин) Ниже приведён алгоритм. Укажите наименьшее число  $x$ , при вводе которого алгоритм напечатает число, большее, чем 10300.

```
#include <iostream>
using namespace std;
int main()
{
 int x, d, x0, N = 0;
 cin >> x;
 x0 = x;
 while (x > 0) {
 d = x%4;
 N = 10*N + d;
 x = x/4;
 }
 N = N + x0;
}
```

```
 cout << N;
 return 0;
}
```

- 99) Ниже приведён алгоритм. Укажите наибольшее число  $x$ , при вводе которого алгоритм напечатает сначала 3, потом 5.

```
#include <iostream>
using namespace std;
int main()
{
 int x, L, M, Q;
 cin >> x;
 Q = 6;
 L = 0;
 while (x >= Q) {
 L = L + 1;
 x = x - Q;
 }
 M = x;
 if(M < L) {
 M = L;
 L = x;
 }
 cout << L << endl << M;
 return 0;
}
```

- 100) Ниже приведён алгоритм. Укажите наибольшее число  $x$ , при вводе которого алгоритм напечатает сначала 8, потом 11.

```
#include <iostream>
using namespace std;
int main()
{
 int x, L, M, Q;
 cin >> x;
 Q = 16;
 L = 0;
 while (x >= Q) {
 L = L + 1;
 x = x - Q;
 }
 M = x;
 if(M < L) {
 M = L;
 L = x;
 }
 cout << L << endl << M;
 return 0;
}
```

- 101) (Д.В. Богданов) Укажите наименьшее натуральное число  $x$ , при вводе которого будет два раза напечатано число 3.

```
#include <iostream>
using namespace std;
```



---

```
int main()
{ int x;
 cin >> x;
 int a = 0, b = 1;
 while (x > 0) {
 if (x % 3 > 0) a++;
 if (x % 3 > 1) b++;
 x /= 10;
 }
 cout << a << endl << b;
 return 0;
}
```

- 102) Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 24?

```
#include <iostream>
using namespace std;
int main()
{ int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0){
 a = a+1;
 b = b*(x%10);
 x = x/10;
 }
 cout << a << endl << b;
 return 0;
}
```

- 103) Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 0?

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0){
 a = a+1;
 b = b*(x%10);
 x = x/10;
 }
 cout << a << endl << b;
 return 0;
}
```

- 104) (Д.Ф. Муфаззалов) Укажите наибольшее двузначное натуральное число, при вводе которого приведенная ниже программа напечатает число 0.

```
#include <iostream>
using namespace std;
int main()
{
```

```
int i, n;
cin >> x;
i = 0;
while (n > 0) {
 i = i + n % 16;
 n = n / 16;
}
cout << i % 15;
return 0;
}
```

- 105) (Д.Ф. Муфаззалов) Укажите наименьшее трехзначное натуральное число, при вводе которого приведенная ниже программа напечатает число 0.

```
#include <iostream>
using namespace std;
int main()
{
 int i, n;
 cin >> n;
 i = 0;
 while (n > 0) {
 i = i + n % 20;
 n = n / 20;
 }
 cout << i % 19;
 return 0;
}
```

- 106) Укажите количество двузначных натуральных чисел, при вводе которых приведенная ниже программа напечатает число 0.

```
#include <iostream>
using namespace std;
int main()
{
 int i, n;
 cin >> n;
 i = 0;
 while (n > 0) {
 i = i + n % 8;
 n = n / 8;
 }
 cout << i % 7;
 return 0;
}
```

- 107) (Д.Ф. Муфаззалов) Укажите количество двузначных натуральных чисел, при вводе которых приведенная ниже программа напечатает число, отличное от 0.

```
#include <iostream>
using namespace std;
int main()
{
 int i, n;
 cin >> n;
 i = 0;
```

```
while (n > 0) {
 i = i + n % 9;
 n = n / 9;
}
cout << i % 8;
return 0;
}
```

- 108) Укажите наименьшее натуральное число, при вводе которого эта программа напечатает сначала 1, потом 4.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 0;
 while (x > 0) {
 if (x%2 == 0) a += 1;
 else b += x%6;
 x = x / 6;
 }
 cout << a << endl << b;
 return 0;
}
```

- 109) Укажите наименьшее натуральное число, при вводе которого эта программа напечатает сначала 2, потом 6.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 0;
 while (x > 0) {
 if (x%2 == 0) a += 1;
 else b += x%6;
 x = x / 6;
 }
 cout << a << endl << b;
 return 0;
}
```

- 110) Укажите наименьшее натуральное число, при вводе которого эта программа напечатает сначала 1, потом 5.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 0;
 while (x > 0) {
```

---

```

 if (x%2 == 0) a += 1;
 else b += x%4;
 x = x / 4;
 }
 cout << a << endl << b;
 return 0;
}

```

- 111) Укажите наименьшее натуральное число, при вводе которого эта программа напечатает сначала 2, потом 7.

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 0;
 while (x > 0) {
 if (x%2 == 0) a += 1;
 else b += x%4;
 x = x / 4;
 }
 cout << a << endl << b;
 return 0;
}

```

- 112) Укажите наибольшее натуральное число, при вводе которого эта программа напечатает сначала 3, потом 6.

```

#include <iostream>
using namespace std;
int main()
{
 int L, M, x;
 cin >> x;
 L = 0; M = 0;
 while (x > 0) {
 L = L + 1;
 if (x % 8 != 0)
 M = M + x % 8;
 x = x / 8;
 }
 cout << L << " " << M;
}

```

- 113) Укажите наибольшее натуральное число, при вводе которого эта программа напечатает сначала 14, потом 3.

```

#include <iostream>
using namespace std;
int main(){
 int x, L, M;
 cin >> x;
 L = 0;
 M = 0;
 while (x > 0) {

```

```

 M = M + 1;
 if(x % 2 != 0)
 L = L + x % 8;
 x = x / 8;
}
cout << L << endl << M;
return 0;
}

```

- 114) Укажите наибольшее трёхзначное натуральное число, при вводе которого эта программа напечатает сначала 2, потом 8.

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 if (x%2 > 0) a += x%8;
 else b *= x%8;
 x = x / 8;
 }
 cout << a << endl << b;
 return 0;
}

```

- 115) Укажите наименьшее натуральное число, при вводе которого эта программа напечатает сначала 2, потом 24.

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 if (x%2 > 0) a += x%8;
 else b *= x%8;
 x = x / 8;
 }
 cout << a << endl << b;
 return 0;
}

```

- 116) Укажите наибольшее трёхзначное натуральное число, при вводе которого эта программа напечатает сначала 2, потом 7.

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;

```

```
while (x > 0) {
 if (x%2 > 0) a += x%6;
 else b += x%6;
 x = x / 6;
}
cout << a << endl << b;
return 0;
}
```

- 117) Укажите наименьшее трёхзначное натуральное число, при вводе которого эта программа напечатает сначала 2, потом 9.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 if (x%2 > 0) a += x%6;
 else b += x%6;
 x = x / 6;
 }
 cout << a << endl << b;
 return 0;
}
```

- 118) Укажите наибольшее трёхзначное натуральное число, при вводе которого эта программа напечатает сначала 2, потом 6.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 if (x%2 > 0) a += 1;
 else b += x%5;
 x = x / 5;
 }
 cout << a << endl << b;
 return 0;
}
```

- 119) Укажите наименьшее трёхзначное натуральное число, при вводе которого эта программа напечатает сначала 2, потом 9.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
```

---

```

while (x > 0) {
 if (x%2 > 0) a += 1;
 else b += x%5;
 x = x / 5;
}
cout << a << endl << b;
return 0;
}

```

- 120) (Д.Ф. Муфаззалов, Уфа) Укажите наименьшее четырехзначное натуральное число, при вводе которого приведенная ниже программа напечатает число 4.

```

#include <iostream>
using namespace std;
int main()
{
 int n, a;
 cin >> n;
 a = -1;
 while (n > 9 && a != n%10) {
 a = n%10;
 n /= 10;
 }
 cout << n%10;
 return 0;
}

```

- 121) (Д.Ф. Муфаззалов, Уфа) Укажите минимальное натуральное число, при вводе которого приведенная ниже программа напечатает число 4, а потом число 7.

```

#include <iostream>
using namespace std;
int main()
{
 int n,a,k;
 cin >> n;
 a = -1; k = 0;
 while (n > 9 && a != n%10) {
 a = n%10;
 n /= 10;
 k++;
 }
 cout << k << endl << a;
 return 0;
}

```

- 122) (Д.Ф. Муфаззалов, Уфа) Укажите минимальное трехзначное натуральное число, при вводе которого приведенная ниже программа напечатает число 5.

```

#include <iostream>
using namespace std;
int main()
{
 int n, a, k;
 cin >> n;
 a = -1;
 while (n > 7 && a != n%8) {

```

```

 a = n%8;
 n /= 8;
}
if (a==n%8)
 cout << a;
else
 cout << n;
return 0;
}

```

- 123) (Д.Ф. Муфаззалов, Уфа) Укажите наибольшее трехзначное натуральное число, при вводе которого приведенная ниже программа напечатает число 5.

```

#include <iostream>
using namespace std;
int main()
{
 int n,a,k;
 cin >> n;
 a = -1;
 while (n > 9 && a != n%10) {
 a = n%10;
 n /= 10;
 }
 if (a == n%10)
 cout << a;
 else
 cout << n;
 return 0;
}

```

- 124) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 4, а потом 7.

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 if (x%2 > 0) a += x%13;
 else b *= x%13;
 x = x / 13;
 }
 cout << a << endl << b;
 return 0;
}

```

- 125) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 5, а потом 2.

```

#include <iostream>
using namespace std;

```



```
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 if (x%2 > 0) a += x%13;
 else b *= x%13;
 x = x / 13;
 }
 cout << a << endl << b;
 return 0;
}
```

- 126) **(С.С. Поляков, Саратов)** Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 2, а потом 9.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 if (x%2 > 0) a += x%11;
 else b *= x%11;
 x = x / 11;
 }
 cout << a << endl << b;
 return 0;
}
```

- 127) **(С.С. Поляков, Саратов)** Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которого алгоритм печатает сначала 1, а потом 8.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 if (x%2 > 0) a += x%11;
 else b *= x%11;
 x = x / 11;
 }
 cout << a << endl << b;
 return 0;
}
```

}

- 128) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наибольшее трёхзначное натуральное число, при вводе которого алгоритм печатает сначала 1, а потом 8.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 if (x%2 > 0) a += x%11;
 else b *= x%11;
 x = x / 11;
 }
 cout << a << endl << b;
 return 0;
}
```

- 129) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 1, а потом 9.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 if (x%2 == 0) a += x%9;
 else b *= x%9;
 x = x / 9;
 }
 cout << a << endl << b;
 return 0;
}
```

- 130) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 3, а потом тоже - 3.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
```

```
while (x > 0) {
 if (x%2 > 0) a += x%7;
 else b *= x%7;
 x = x / 7;
}
cout << a << endl << b;
return 0;
}
```

- 131) **(С.С. Поляков, Саратов)** Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 3, а потом 12.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 if (x%2 > 0) a += x%13;
 else b *= x%13;
 x = x / 13;
 }
 cout << a << endl << b;
 return 0;
}
```

- 132) **(С.С. Поляков, Саратов)** Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее четырёхзначное натуральное число, при вводе которого алгоритм печатает сначала 4, а потом тоже - 4.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 0;
 while (x > 0) {
 if (x%2 > 0)
 a += x%9;
 else
 b += x%9;
 x = x / 9;
 }
 cout << a << endl << b;
 return 0;
}
```

- 133) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наибольшее трёхзначное натуральное число, при вводе которого алгоритм печатает сначала 4, а потом тоже - 4.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 0;
 while (x > 0) {
 if (x%2 > 0)
 a += x%5;
 else
 b += x%5;
 x = x / 5;
 }
 cout << a << endl << b;
 return 0;
}
```

- 134) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 0?

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 a += 1;
 b *= x%10;
 x = x / 10;
 }
 cout << a << endl << b;
 return 0;
}
```

- 135) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 5?

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0;
```

```

while (x > 0) {
 a += 1;
 b = x%6;
 x = x / 6;
}
cout << a << endl << b;
return 0;
}

```

- 136) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 5, а потом 7?

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0;
 while (x > 0) {
 a += 1;
 b = x%9;
 x = x / 9;
 }
 cout << a << endl << b;
 return 0;
}

```

- 137) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 24?

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0)
 { a++;
 if (x%8 != 1)
 b *= x%8;
 x = x / 8;
 }
 cout << a << endl << b;
 return 0;
}

```

- 138) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 24?

```

#include <iostream>

```

---

```
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0)
 {
 a++;
 if (x%8 != 0)
 b *= x%8;
 x = x / 8;
 }
 cout << a << endl << b;
 return 0;
}
```

- 139) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 12?

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0)
 {
 a++;
 if (x%14 != 0)
 b *= x%14;
 x = x / 14;
 }
 cout << a << endl << b;
 return 0;
}
```

- 140) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 10?

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0; b = 1;
 while (x > 0)
 {
 a++;
 if (x%12 != 0)
 b *= x%12;
 x = x / 12;
 }
 cout << a << endl << b;
```

```
 return 0;
```

```
}
```

- 141) (Д. Ф. Муфаззалов, Уфа) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает число, сумма цифр которого равна 15.

```
#include <iostream>
using namespace std;
int main()
{
 int i, x, ans=0;
 cin >> x;
 i = 1;
 while (x > 0) {
 ans = x % 10*i + ans;
 x = x / 10;
 i = i*10;
 }
 cout << ans;
 return 0;
}
```

- 142) (Д. Ф. Муфаззалов, Уфа) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает число, сумма цифр которого равна 15.

```
#include <iostream>
using namespace std;
int main()
{
 int i, x, ans=0;
 cin >> x;
 i = 1;
 while (x > 0) {
 ans = x % 8*i + ans;
 x = x / 8;
 i = i*10;
 }
 cout << ans;
 return 0;
}
```

- 143) (Д. Ф. Муфаззалов, Уфа) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает число 110.

```
#include <iostream>
using namespace std;
int main()
{
 int i, x, ans=0;
 cin >> x;
 i = 1;
 while (x > 0) {
 ans = x % 16*i + ans;
 x = x / 16;
 i = i*10;
 }
 cout << ans;
```

```
 return 0;
}
```

- 144) Укажите минимальное двузначное натуральное число, при вводе которого этот алгоритм напечатает число 8.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 0; b = 10;
 while (x > 0) {
 d = x % 6;
 if (d > a) a = d;
 if (d < b) b = d;
 x = x / 6;
 }
 cout << a+b << endl;
 return 0;
}
```

- 145) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает число 11.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 0; b = 10;
 while (x > 0) {
 d = x % 7;
 if (d > a) a = d;
 if (d < b) b = d;
 x = x / 7;
 }
 cout << a+b << endl;
 return 0;
}
```

- 146) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает сначала 7, а потом 12.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 0; b = 1;
 while (x > 0) {
 if (x % 2 > 0)
 a += x % 12;
 else b *= x % 12;
 }
}
```



```

 x = x / 12;
 }
 cout << a << endl;
 cout << b << endl;
 return 0;
}

```

- 147) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает число 18.

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 0; b = 10;
 while (x > 0) {
 d = x % 9;
 if (d > a) a = d;
 if (d < b) b = d;
 x = x / 9;
 }
 cout << a*b << endl;
 return 0;
}

```

- 148) (Е. Джобс) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , большее, чем 100, при вводе которого алгоритм напечатает 21.

```

#include <iostream>
using namespace std;
int main()
{
 int x, L, M;
 cin >> x;
 L = 3*x - 6;
 M = 3*x + 99;
 while(L != M)
 if(L > M)
 L = L - M;
 else
 M = M - L;
 cout << M << endl;
}

```

- 149) (Е. Джобс) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наибольшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 2, а потом 1.

```

#include <iostream>
using namespace std;
int main()
{
 int x, y, a, b;
 cin >> x;
 a = 0; b = 0;
}

```

```

while(x > 0) {
 y = x % 10;
 if(y > 3)
 a = a + 1;
 else
 b = b - 1;
 if(y < 8)
 b = b + 1;
 x = x / 10;
}
cout << a << endl;
cout << b << endl;
}

```

- 150) (Е. Джобс) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наибольшее натуральное число, при вводе которого алгоритм печатает сначала 13, а потом 3.

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 0;
 b = 0;
 while(x > 0) {
 x = x / 9;
 if(x % 2 > 0)
 a = a + x % 9;
 b = b + 1;
 }
 cout << a << endl;
 cout << b << endl;
}

```

- 151) (Досрочный ЕГЭ-2020) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $L$  и  $M$ . Укажите наибольшее натуральное число, при вводе которого алгоритм печатает сначала 2, а потом 3.

```

#include <iostream>
using namespace std;
int main(){
 int x, L, M;
 cin >> x;
 L = 0;
 M = 0;
 while (x > 0) {
 M = M + 1;
 if(x % 2 != 0) {
 L = L + x % 8;
 }
 x = x / 8;
 }
 cout << L << endl << M << endl;
}

```

```
return 0;
}
```

- 152) (Досрочный ЕГЭ-2020) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $L$  и  $M$ . Укажите наибольшее натуральное число, при вводе которого алгоритм печатает сначала 12, а потом 3.

```
#include <iostream>
using namespace std;
int main() {
 int x, L, M;
 cin >> x;
 L = 0;
 M = 0;
 while (x > 0) {
 M = M + 1;
 if(x % 2 == 0) {
 L = L + x % 8;
 }
 x = x / 8;
 }
 cout << L << endl << M << endl;
 return 0;
}
```

- 153) (А.М. Кабанов) Ниже приведён алгоритм. Получив на вход число  $x$ , этот алгоритм печатает число  $S$ . Укажите наибольшее число  $x$ , при вводе которого алгоритм печатает 82.

```
#include<iostream>
using namespace std;

int main() {
 int x, S, P, K;
 cin >> x;
 P = 90;
 S = 6 * (x - x % 22);
 K = 0;
 while (P < 181) {
 K = K + 1;
 P = P + K;
 S = S - 2 * K;
 }
 cout << S;
 return 0;
}
```

- 154) (А.М. Кабанов) Ниже приведён алгоритм. Получив на вход число  $x$ , этот алгоритм печатает число  $K$ . Укажите наименьшее число  $x$ , при вводе которого алгоритм печатает 10.

```
#include<iostream>
using namespace std;

int main() {
 int x, S, P, K;
 cin >> x;
 P = x;
 S = 10 * x;
 K = 0;
```

```

while (P < S) {
 K = K + 1;
 S = S - 2 * K;
 P = P + K;
}
cout << K;
return 0;
}

```

- 155) (А.М. Кабанов) Ниже приведён алгоритм. Получив на вход число  $x$ , этот алгоритм печатает число  $K$ . Укажите наибольшее число  $x$ , при вводе которого алгоритм печатает 5.

```

#include<iostream>
using namespace std;

int main() {
 int x, S, P, K;
 cin >> x;
 K = x - 1;
 P = 100;
 S = 340;
 while (P < S) {
 K = K + 1;
 S = S - 2 * K;
 P = P + K;
 }
 K = K - x;
 cout << K;
}

```

- 156) (Т.Л. Шамасова) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 11, а потом 3.

```

#include <iostream>
using namespace std;
int main() {
 int x, a, b;
 cin >> x;
 a = 0; b = 0;
 while(x > 0) {
 x = x / 9;
 if(x % 2 > 0)
 a = a + x % 9;
 else b = b + 1;
 }
 cout << a << endl;
 cout << b;
}

```

- 157) Получив на вход натуральное число  $x$ , этот алгоритм печатает одно число. Укажите наименьшее натуральное число, при вводе которого алгоритм печатает 48.

```

#include <stdio.h>
void main()
{
 int x, a;
 cin >> x;
 a = 1;
 while (x > 0) {

```

```

 a *= x % 7;
 x = x / 7;
}
cout << a << endl;
}

```

- 158) Получив на вход натуральное число  $x$ , этот алгоритм печатает одно число. Укажите наименьшее натуральное число, при вводе которого алгоритм печатает 54.

```

#include <stdio.h>
void main()
{
 int x, a;
 cin >> x;
 a = 1;
 while (x > 0) {
 a *= x % 7;
 x = x / 7;
 }
 cout << a << endl;
}

```

- 159) Получив на вход натуральное число  $x$ , этот алгоритм печатает одно число. Укажите наименьшее натуральное число, при вводе которого алгоритм печатает 40.

```

#include <stdio.h>
void main()
{
 int x, a;
 cin >> x;
 a = 1;
 while (x > 0) {
 a *= x % 6;
 x = x / 6;
 }
 cout << a << endl;
}

```

- 160) Получив на вход натуральное число  $x$ , этот алгоритм печатает одно число. Укажите наименьшее натуральное число, при вводе которого алгоритм печатает 60.

```

#include <stdio.h>
void main()
{
 int x, a;
 cin >> x;
 a = 1;
 while (x > 0) {
 a *= x % 9;
 x = x / 9;
 }
 cout << a << endl;
}

```

- 161) (Е. Джобс) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 2.

```

#include <iostream>
using namespace std;

int main() {

```

```

int x, L, M;
cin >> x;
L = 1;
M = 0;
while(x > 0){
 if(M < L)
 M = M + 1;
 else
 L = L + M;
 x = x / 6;
}
cout << L << endl << M;
}

```

- 162) (Е. Джобс) Ниже записана программа. Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает два числа:  $L$  и  $M$ . Укажите наибольшее число  $x$ , при вводе которого алгоритм выводит сначала 2, а потом 4.

```

#include <iostream>
using namespace std;
int main() {
 int x, L, M;
 cin >> x;
 L = 0;
 M = 10;
 while (x > 5) {
 L = L + 1;
 if(x % 8 < M)
 M = x % 8;
 x = x / 8;
 }
 cout << L << endl << M;
}

```

- 163) (Е. Джобс) Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает число  $S$ . Какое наименьшее число  $x$  необходимо ввести, чтобы в результате работы алгоритма на экран было выведено число, большее 100?

```

#include <iostream>
using namespace std;
int main() {
 int x, A, B, S;
 cin >> x;
 B = x; S = -2; A = 4;
 while (B / 2 > 0) {
 if(B % 2 == 0)
 S = S + A;
 else
 S = S * 3;
 B = B / 2;
 }
 cout << S << endl;
}

```

- 164) (Е. Джобс) Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает два числа. Укажите наименьшее число, при вводе которого программа напечатает сначала 3, затем 4.

```

#include <iostream>
using namespace std;

```

```

int main() {
 int x, A, B;
 A = 0;
 B = 1;
 cin >> x;
 while (x > 0) {
 if(x % 7 > 2)
 A = A + 1;
 else
 B = B * (x % 7);
 x = x / 7;
 }
 cout << A << endl;
 cout << B << endl;
 return 0;
}

```

- 165) (Е. Джобс) Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает два числа. Укажите наибольшее число, при вводе которого программа напечатает сначала 270, затем 90.

```

#include <iostream>
using namespace std;
int main() {
 int x, P, S, i;
 cin >> x;
 P = 0;
 S = 10*(x - x % 15);
 i = 2;
 while (i < 20) {
 S = S - 2 * i;
 P = P + i;
 i = i + 2;
 }
 cout << S << endl;
 cout << P << endl;
}

```

- 166) (Е. Джобс) Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает число  $S$ . Укажите наименьшее число  $x$ , при вводе которого на экран будет выведено число, большее 100.

```

#include <iostream>
using namespace std;
int main() {
 int x, S, A;
 S = 1;
 A = 5;
 while (x / 7 > 0) {
 if(x % 2 == 0)
 S = S + A;
 else
 S = S * (x % 7);
 x = x / 7;
 }
 cout << S;
}

```

- 167) (Е. Джобс) Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает число  $S$ . Известно, что в результате работы программы на экран выведено минимально возможное число большее 25. Укажите минимальное число  $x$ , для которого это возможно.

```
#include <iostream>
using namespace std;
int main() {
 int x, S, A;
 cin >> x;
 S = 1;
 A = 11;
 while (x / 7 > 0) {
 if(x % 7 < 4)
 S = S + A;
 else
 S = S + (x % 7);
 x = x / 7;
 }
 cout << S;
 return 0;
}
```

- 168) (Е. Джобс) Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает число  $S$ . Укажите наименьшее число  $x$ , при вводе которого на экран будет выведено натуральное число, кратное 100.

```
#include <iostream>
using namespace std;
int main() {
 int x, S;
 cin >> x;
 S = 5;
 while (x > 0) {
 if(x % 8 > 4)
 S = S + (x % 8);
 else
 S = S * (x % 8);
 x = x / 8;
 }
 cout << S;
 return 0;
}
```

- 169) (Е. Джобс) Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает число  $S$ . Сколько существует чисел  $x$ , не превышающих 500, при вводе которых результате работы программы на экране будет выведено число 13.

```
#include <iostream>
using namespace std;
int main() {
 int x, S;
 cin >> x;
 S = 0;
 while (x > 0) {
 if(x % 5 > 0)
 S = S + (x % 5);
 else
 S = S * (x % 5);
 x = x / 5;
 }
```



```

 }
 cout << S;
 return 0;
}

```

- 170) (Е. Джобс) Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает число  $S$ . Укажите наименьшее число  $x$ , большее 50, при вводе которого на экран будет выведено число 1.

```

#include <iostream>
using namespace std;
int main() {
 int x, S;
 cin >> x;
 S = 0;
 while (x > 0) {
 if (x % 2 > 0)
 S = S + (x % 7);
 else
 S = S - (x % 7);
 x = x / 7;
 }
 cout << S;
 return 0;
}

```

- 171) (А. Кабанов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа. Укажите наибольшее натуральное число, при вводе которого алгоритм печатает сначала 24, а затем 4.

```

#include <stdio.h>
using namespace std;
int main() {
 int x, a, b;
 cin >> x;
 a = 0;
 b = 0;
 while (x > 0) {
 a = a + 1;
 if (x % 2 != 0)
 b = b + 1;
 x = x / 2;
 }
 cout << a << b << endl;
}

```

- 172) (А. Кабанов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа. Укажите наибольшее натуральное число, при вводе которого алгоритм печатает сначала 10, а затем 8.

```

#include <stdio.h>
using namespace std;
int main() {
 int x, a, b;
 cin >> x;
 a = 0;
 b = 0;
 while (x > 0) {
 if (x % 2 == 0)
 a = a + 1;
 }
}

```

```

 else
 b = b + 1;
 x = x / 2;
 }
 cout << a << b << endl;
}

```

173) (А. Кабанов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа.

Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 7, а затем 7.

```

#include <iostream>
using namespace std;
int main() {
 int x, a, b;
 cin >> x;
 a = 0;
 b = 0;
 while (x > 0) {
 a = a + 1;
 if (x % 11 > b)
 b = x % 11;
 x = x / 11;
 }
 cout << a << b << endl;
}

```

174) (А. Кабанов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа.

Укажите наименьшие натуральные значения  $x$  и  $y$ , при вводе которого алгоритм печатает сначала 6, а затем 7. В качестве ответа запишите произведение  $x \cdot y$ .

```

#include <iostream>
using namespace std;
int main() {
 int x, y, a, b;
 cin >> x;
 cin >> y;
 a = 0;
 b = 0;
 while (x > 0 || y > 0) {
 if (x > 0) a = a + 1;
 if (y > 0) b = b + 1;
 x = x / 2;
 y = y / 10;
 }
 cout << a << b << endl;
}

```

175) (А. Кабанов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа.

Укажите наименьшие натуральные значения  $x$  и  $y$ , при вводе которого алгоритм печатает сначала 4, а затем 5. В качестве ответа запишите произведение  $x \cdot y$ .

```

#include <iostream>
using namespace std;
int main() {
 int x, y, a, b;
 cin >> x;
 cin >> y;
 a = 0;
 b = 0;
}

```

```

while (x * y > 0) {
 if (x > 0) a = a + 1;
 if (y > 0 && y % 7 > b)
 b = y % 7;
 x = x / 10;
 y = y / 7;
}
cout << a << b << endl;
}

```

- 176) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Укажите наименьшее возможное значение  $x$ , при вводе которого программа выведет числа 5 и 12.

```

#include <iostream>
using namespace std;
int main() {
 int x, m, s, d;
 cin >> x;
 m = 0;
 s = 0;
 while (x > 0) {
 d = x % 7;
 s += d;
 if (d > m) m = d;
 x = x / 7;
 }
 cout << m << ' ' << s << endl;
 return 0;
}

```

- 177) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Укажите наименьшее возможное значение  $x$ , при вводе которого программа выведет числа 3 и 14.

```

#include <iostream>
using namespace std;
int main() {
 int x, m, s, d;
 cin >> x;
 m = 0;
 s = 0;
 while (x > 0) {
 d = x % 7;
 s += d;
 if (d > m) m = d;
 x = x / 7;
 }
 cout << m << ' ' << s << endl;
 return 0;
}

```

- 178) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Укажите наименьшее возможное значение  $x$ , при вводе которого программа выведет числа 3 и 10.

```

#include <iostream>
using namespace std;
int main() {
 int x, m, s, d;

```

```

cin >> x;
m = 0;
s = 0;
while (x > 0) {
 d = x % 6;
 s += d;
 if (d > m) m = d;
 x = x / 6;
}
cout << m << ' ' << s << endl;
return 0;
}

```

- 179) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Укажите наименьшее возможное значение  $x$ , при вводе которого программа выведет числа 5 и 16.

```

#include <iostream>
using namespace std;
int main() {
 int x, m, s, d;
 cin >> x;
 m = 0;
 s = 0;
 while (x > 0) {
 d = x % 6;
 s += d;
 if (d > m) m = d;
 x = x / 6;
 }
 cout << m << ' ' << s << endl;
 return 0;
}

```

- 180) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Укажите наименьшее возможное значение  $x$ , при вводе которого программа выведет числа 3 и 10.

```

#include <iostream>
using namespace std;
int main()
{
 int x, k, a, b, d;
 cin >> x;
 k = x % 5;
 a = 0; b = 0;
 while (x > 0) {
 d = x % 5;
 if (d == k) ++a;
 b += d;
 x = x / 5;
 }
 cout << a << ' ' << b << endl;
 return 0;
}

```

- 181) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Укажите наименьшее возможное значение  $x$ , при вводе которого программа выведет числа 2 и 15.

```
#include <iostream>
using namespace std;
int main()
{
 int x, k, a, b, d;
 cin >> x;
 k = x % 6;
 a = 0; b = 0;
 while (x > 0) {
 d = x % 6;
 if (d == k) ++a;
 b += d;
 x = x / 6;
 }
 cout << a << ' ' << b << endl;
 return 0;
}
```

- 182) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Укажите наименьшее возможное значение  $x$ , при вводе которого программа выведет числа 4 и 11.

```
#include <iostream>
using namespace std;
int main()
{
 int x, k, a, b, d;
 cin >> x;
 k = x % 7;
 a = 0; b = 0;
 while (x > 0) {
 d = x % 7;
 if (d == k) ++a;
 b += d;
 x = x / 7;
 }
 cout << a << ' ' << b << endl;
 return 0;
}
```

- 183) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Укажите наименьшее возможное значение  $x$ , при вводе которого программа выведет числа 3 и 20.

```
#include <iostream>
using namespace std;
int main()
{
 int x, k, a, b, d;
 cin >> x;
 k = x % 8;
 a = 0; b = 0;
 while (x > 0) {
 d = x % 8;
 if (d == k) ++a;
 b += d;
 x = x / 8;
 }
}
```

```
cout << a << ' ' << b << endl;
return 0;
}
```

- 184) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит результат. Укажите наименьшее значение  $x$ , при вводе которого программа выведет число 64.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 3*x + 67;
 b = 3*x - 61;
 while (a != b) {
 if (a > b)
 a -= b;
 else
 b -= a;
 }
 cout << a << endl;
 return 0;
}
```

- 185) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит результат. Укажите наименьшее значение  $x$ , при вводе которого программа выведет число 96.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 5*x + 345;
 b = 5*x - 807;
 while (a != b) {
 if (a > b)
 a -= b;
 else
 b -= a;
 }
 cout << a << endl;
 return 0;
}
```

- 186) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит результат. Укажите наименьшее значение  $x$ , при вводе которого программа выведет число 158.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 3*x + 71;
 b = 3*x - 87;
```

```
while (a != b) {
 if (a > b)
 a -= b;
 else
 b -= a;
}
cout << a << endl;
return 0;
}
```

- 187) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит результат. Укажите наименьшее значение  $x$ , при вводе которого программа выведет число 34.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = 3*x - 112;
 b = 3*x + 58;
 while (a != b) {
 if (a > b)
 a -= b;
 else
 b -= a;
 }
 cout << a << endl;
 return 0;
}
```

- 188) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит результат. Укажите наименьшее значение  $x$ , при вводе которого программа выведет число 45.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b;
 cin >> x;
 a = x - 61;
 b = 3*x - 138;
 while (a != b) {
 if (a > b)
 a -= b;
 else
 b -= a;
 }
 cout << a << endl;
 return 0;
}
```

- 189) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит результат. Укажите наименьшее значение  $x$ , при вводе которого программа выведет число 15.

```
#include <iostream>
using namespace std;
```

```

int main()
{
 int x, a, b;
 cin >> x;
 a = 2*x - 91;
 b = 3*x - 159;
 while (a != b) {
 if (a > b)
 a -= b;
 else
 b -= a;
 }
 cout << a << endl;
 return 0;
}

```

- 190) (Е. Джобс) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа  $L$  и  $M$ . Сколько существует натуральных чисел  $x$ , при вводе которых алгоритм печатает 6 и 0?

```

#include <iostream>
using namespace std;
int main(){
 int L = 0, M = 0, x;
 cin >> x;
 while (x > 0) {
 L = L+1;
 if (x % 16 % 2 == 0)
 M = M + 1;
 else
 M = M - 1;
 x = x / 16;
 }
 cout << L << endl << M;
 return 0;
}

```

- 191) (Е. Джобс) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует натуральных чисел, цифры в которых расположены в порядке невозрастания, при вводе которых алгоритм печатает сначала 9, а потом 5.

```

#include <iostream>
using namespace std;
int main() {
 int x, a, b;
 cin >> x;
 x = (x - x % 8) * 10;
 a = 1; b = 0;
 while (x > 0) {
 if (x%2 != 0)
 a *= x%4;
 else
 b += x % 4;
 x = x / 8;
 }
 cout << a << endl << b;
 return 0;
}

```



- 192) (Е. Джобс) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа  $L$  и  $M$ . При каком наибольшем значении  $x$  после выполнения программы на экран будет выведено сначала число 3, а затем – 7.

```
#include <iostream>
using namespace std;
int main () {
 int L = 0, M = 0, x;
 cin >> x;
 while (x > 12) {
 L = L + 1;
 x = x / 4;
 }
 M = x;
 if(L > M) {
 int t = L;
 L = M; M = t;
 }
 cout << L << endl << M;
 return 0;
}
```

- 193) (А. Богданов) Ниже записан алгоритм. Получив на вход число  $X$ , этот алгоритм печатает одно число. Укажите наименьшее число  $X$ , большее 80, при вводе которого алгоритм печатает число 17.

```
#include <iostream>
using namespace std;
int main() {
 int x, s = 0;
 cin >> x;
 s = 0;
 while (x > 0) {
 s = s + x % 9;
 x = x / 3;
 }
 cout << s << endl;
 return 0;
}
```

- 194) (А. Богданов) Ниже записан алгоритм. Получив на вход число  $K$ , этот алгоритм печатает два числа. Укажите наименьшее число  $K$ , при вводе которого алгоритм печатает числа 13 и 25.

```
#include <iostream>
using namespace std;
int main() {
 int k, t, x, y;
 cin >> k;
 x = 1; y = x;
 while (k < 13) {
 k += 1;
 if (k == 7)
 t = 0;
 else
 t = x + y;
 x = y; y = t;
 }
 cout << k << ' ' << x;
```

```
 return 0;
}
```

- 195) (А. Кабанов) Ниже записана программа. Получив на вход число N, эта программа печатает число с. При каком наибольшем натуральном значении переменной N программа выведет 9?

```
#include <iostream>
using namespace std;
int main() {
 int c = 0, T = 3, d = 3, N;
 cin >> N;
 while (N != 0) {
 N = N - T;
 T = T + d;
 c = c + 1;
 }
 if (c % 2 == 0) c = c + d;
 cout << c;
 return 0;
}
```

- 196) (А. Кабанов) Ниже записана программа. Получив на вход число N, эта программа печатает число с. При каком наименьшем натуральном значении переменной N программа выведет 8?

```
#include <iostream>
using namespace std;
int main() {
 int c = 0, T = 3, d = 3, N;
 cin >> N;
 while (N != 0) {
 N = N - T;
 T = T + d;
 c = c + 1;
 }
 if (c % 2 != 0) c = c + d;
 cout << c;
 return 0;
}
```

- 197) (А. Кабанов) Ниже записана программа. Получив на вход число N, эта программа печатает число с. При каком наименьшем натуральном значении переменной N программа выведет 6?

```
#include <iostream>
using namespace std;
int main() {
 int c = 0, T = 5, d = 5, N;
 cin >> N;
 while (N != 200) {
 N = N + T;
 T = T + d;
 c = c + 1;
 }
 if (c % 2 != 0) c = c + d;
 cout << c;
 return 0;
}
```

- 198) (А. Кабанов) Ниже записана программа. Получив на вход число Т, эта программа печатает число с. При каком наименьшем натуральном значении переменной Т программа выведет 8?

```
#include <iostream>
using namespace std;
int main() {
 int c = 0, N = 0, d = 5, T;
 cin >> T;
 while (N != 300) {
 N = N + T;
 T = T + d;
 c = c + 1;
 }
 if (c % 2 != 0) c = c + d;
 cout << c;
 return 0;
}
```

- 199) (А. Кабанов) Ниже записана программа. Получив на вход число d, эта программа печатает число с. При каком наибольшем натуральном значении переменной d программа выведет 8?

```
#include <iostream>
using namespace std;
int main() {
 int c = 0, N = 0, d, T;
 cin >> d;
 T = d;
 while (N != 144) {
 N = N + T;
 T = T + d;
 c = c + 1;
 }
 if (c % 2 != 0) c = c + 5;
 cout << c;
 return 0;
}
```

- 200) Ниже записана программа. Получив на вход число x, эта программа печатает числа a и b. При каком наибольшем натуральном значении переменной x программа выведет сначала 10, а потом 13?

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 1;
 b = 0;
 while (x > 0) {
 d = x % 9;
 a *= d;
 b += d;
 x /= 9;
 }
 cout << a << ' ' << b << endl;
 return 0;
}
```

```
}
```

- 201) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 10, а потом 14?

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 1;
 b = 0;
 while (x > 0) {
 d = x % 8;
 a *= d;
 b += d;
 x /= 8;
 }
 cout << a << ' ' << b << endl;
 return 0;
}
```

- 202) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 12, а потом 14?

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 1;
 b = 3;
 while (x > 0) {
 d = x % 7;
 a *= d;
 if (d < 4)
 b += d;
 x /= 7;
 }
 cout << a << ' ' << b << endl;
 return 0;
}
```

- 203) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 30, а потом 31?

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 5;
 b = 20;
 while (x > 0) {
```

```

 d = x % 6;
 a *= d;
 if (d < 3)
 b += d;
 x /= 6;
}
cout << a << ' ' << b << endl;
return 0;
}

```

- 204) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 28, а потом 17?

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 4;
 b = 10;
 while (x > 0) {
 d = x % 11;
 a *= d;
 if (d < 5)
 b += d;
 x /= 11;
 }
 cout << a << ' ' << b << endl;
 return 0;
}

```

- 205) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 24, а потом 16?

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 2;
 b = 3;
 while (x > 0) {
 d = x % 4;
 a *= d;
 if (d < 3)
 b += d;
 x /= 4;
 }
 cout << a << ' ' << b << endl;
 return 0;
}

```

- 206) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 63, а потом 15?

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d, w;
 cin >> x;
 a = 3;
 b = 5;
 w = 9;
 while (x > 0) {
 d = x % w;
 a *= d;
 if (d < 5)
 b += d;
 x /= w;
 w = 17 - w;
 }
 cout << a << ' ' << b << endl;
 return 0;
}
```

- 207) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 168, а потом 25?

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d, w;
 cin >> x;
 a = 7;
 b = 15;
 w = 6;
 while (x > 0) {
 d = x % w;
 a *= d;
 if (d < 3)
 b += d;
 x /= w;
 w = 11 - w;
 }
 cout << a << ' ' << b << endl;
 return 0;
}
```

- 208) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 216, а потом 25?

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d, w;
 cin >> x;
 a = 6;
 b = 18;
```

```

w = 9;
while (x > 0) {
 d = x % w;
 a *= d;
 if (d < 3)
 b += d;
 x /= w;
 w = 16 - w;
}
cout << a << ' ' << b << endl;
return 0;
}

```

- 209) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 210, а потом 21?

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d, w;
 cin >> x;
 a = 2;
 b = 14;
 w = 9;
 while (x > 0) {
 d = x % w;
 a *= d;
 if (d < 5)
 b += d;
 x /= w;
 w = 15 - w;
 }
 cout << a << ' ' << b << endl;
 return 0;
}

```

- 210) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 90, а потом 33?

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d, w;
 cin >> x;
 a = 3;
 b = 24;
 w = 8;
 while (x > 0) {
 d = x % w;
 a *= d;
 if (d < 5)
 b += d;
 x /= w;
 w = 11 - w;
 }
}

```

```

 }
 cout << a << ' ' << b << endl;
 return 0;
}

```

- 211) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 225, а потом 27?

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d, w;
 cin >> x;
 a = 5;
 b = 13;
 w = 7;
 while (x > 0) {
 d = x % w;
 a *= d;
 if (d < 5)
 b += d;
 x /= w;
 w = 12 - w;
 }
 cout << a << ' ' << b << endl;
 return 0;
}

```

- 212) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 4, а потом 166?

```

#include <iostream>
using namespace std;
int main()
{
 int x, a = 0, b = 0;
 cin >> x;
 while (x > 0) {
 a += 1;
 if (x % 2 == 0)
 b += x % 100;
 x /= 10;
 }
 cout << a << ' ' << b;
 return 0;
}

```

- 213) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 3, а потом 126?

```

#include <iostream>
using namespace std;
int main()
{
 int x, a = 0, b = 0;
 cin >> x;

```



```

while (x > 0) {
 a += 1;
 if (x % 2 == 1)
 b += x % 100;
 x /= 10;
}
cout << a << ' ' << b;
return 0;
}

```

- 214) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 8, а потом 146?

```

#include <iostream>
using namespace std;
int main()
{
 int x, a = 5, b = 8;
 cin >> x;
 while (x > 0) {
 a += 1;
 if (x % 2 == 1)
 b += x % 100;
 x /= 10;
 }
 cout << a << ' ' << b;
 return 0;
}

```

- 215) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 7, а потом 255?

```

#include <iostream>
using namespace std;
int main()
{
 int x, a = 3, b = 12;
 cin >> x;
 while (x > 0) {
 a += 1;
 if (x % 2 == 1)
 b += x % 100;
 x /= 10;
 }
 cout << a << ' ' << b;
 return 0;
}

```

- 216) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 7, а потом 18?

```

#include <iostream>
using namespace std;
int main()
{
 int x, a = 4, b = 10;
 cin >> x;
}

```

```

while (x > 0) {
 a += 1;
 if (x % 2 == 1)
 b += x % 64;
 x /= 8;
}
cout << a << ' ' << b;
return 0;
}

```

- 217) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 10, а потом 79?

```

#include <iostream>
using namespace std;
int main()
{
 int x, a = 14, b = 100;
 cin >> x;
 while (x > 0) {
 a -= 1;
 if (x % 2 == 1)
 b -= x % 64;
 x /= 8;
 }
 cout << a << ' ' << b;
 return 0;
}

```

- 218) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 58, а потом 415?

```

#include <iostream>
using namespace std;
int main()
{
 int x, a = 62, b = 320;
 cin >> x;
 while (x > 0) {
 a -= 1;
 if (x % 2 == 1)
 b += x % 100;
 x /= 10;
 }
 cout << a << ' ' << b;
 return 0;
}

```

- 219) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 22, а потом 246?

```

#include <iostream>
using namespace std;
int main()
{
 int x, a = 18, b = 432;
 cin >> x;
}

```

```

while (x > 0) {
 a += 1;
 if (x % 2 == 1)
 b -= x % 100;
 x /= 10;
}
cout << a << ' ' << b;
return 0;
}

```

- 220) (Досрочный ЕГЭ-2022) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наименьшем натуральном значении переменной  $s$  программа выведет сначала 10, а потом 19?

```

#include <iostream>
using namespace std;
int main() {
 int P = 10, Q = 8, s;
 int K1 = 0, K2 = 0;
 cin >> s;
 while (s <= 100) {
 s = s + P;
 K1 = K1 + 1;
 }
 while (s >= Q) {
 s = s - Q;
 K2 = K2 + 1;
 }
 K1 = K1 + s; K2 = K2 + s;
 cout << K1 << endl << K2;
 return 0;
}

```

- 221) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наименьшем натуральном значении переменной  $s$  программа выведет сначала 8, а потом 70?

```

#include <iostream>
using namespace std;
int main() {
 int P = 11, Q = 3, s;
 int K1 = 0, K2 = 0;
 cin >> s;
 while (s <= 200) {
 s = s + P;
 K1 = K1 + 1;
 }
 while (s >= Q) {
 s = s - Q;
 K2 = K2 + 1;
 }
 K1 = K1 + s; K2 = K2 + s;
 cout << K1 << endl << K2;
 return 0;
}

```

- 222) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наименьшем натуральном значении переменной  $s$  программа выведет сначала 8, а потом 26?

```

#include <iostream>
using namespace std;
int main() {
 int P = 10, Q = 5, s;
 int K1 = 0, K2 = 0;
 cin >> s;
 while (s <= 120) {
 s = s + P;
 K1 = K1 + 1;
 }
 while (s >= Q) {
 s = s - Q;
 K2 = K2 + 1;
 }
 K1 = K1 + s; K2 = K2 + s;
 cout << K1 << endl << K2;
 return 0;
}

```

- 223) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наибольшем натуральном значении переменной  $s$  программа выведет сначала 9, а потом 24?

```

#include <iostream>
using namespace std;
int main() {
 int P = 12, Q = 9, s;
 int K1 = 0, K2 = 0;
 cin >> s;
 while (s <= 150) {
 s = s + P;
 K1 = K1 + 1;
 }
 while (s >= Q) {
 s = s - Q;
 K2 = K2 + 1;
 }
 K1 = K1 + s; K2 = K2 + s;
 cout << K1 << endl << K2;
 return 0;
}

```

- 224) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наибольшем натуральном значении переменной  $s$  программа выведет сначала 4, а потом 38?

```

#include <iostream>
using namespace std;
int main() {
 int P = 19, Q = 4, s;
 int K1 = 0, K2 = 0;
 cin >> s;
 while (s <= 134) {
 s = s + P;
 K1 = K1 + 1;
 }
 while (s >= Q) {
 s = s - Q;
 }
}

```

```

 K2 = K2 + 1;
 }
 K1 = K1 + s; K2 = K2 + s;
 cout << K1 << endl << K2;
 return 0;
}

```

- 225) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наибольшем натуральном значении переменной  $s$  программа выведет сначала 11, а потом 33?

```

#include <iostream>
using namespace std;
int main() {
 int P = 17, Q = 5, s;
 int K1 = 0, K2 = 0;
 cin >> s;
 while (s <= 140) {
 s = s + P;
 K1 = K1 + 1;
 }
 while (s >= Q) {
 s = s - Q;
 K2 = K2 + 1;
 }
 K1 = K1 + s; K2 = K2 + s;
 cout << K1 << endl << K2;
 return 0;
}

```

- 226) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. Найдите сумму натуральных значений переменной  $s$ , при которых программа выведет сначала 14, а потом 40?

```

#include <iostream>
using namespace std;
int main() {
 int P = 14, Q = 7, s;
 int K1 = 0, K2 = 0;
 cin >> s;
 while (s <= 250) {
 s = s + P;
 K1 = K1 + 1;
 }
 while (s >= Q) {
 s = s - Q;
 K2 = K2 + 1;
 }
 K1 = K1 + s; K2 = K2 + s;
 cout << K1 << endl << K2;
 return 0;
}

```

- 227) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. Найдите сумму натуральных значений переменной  $s$ , при которых программа выведет сначала 12, а потом 75?

```

#include <iostream>
using namespace std;
int main() {

```

```

int P = 15, Q = 3, s;
int K1 = 0, K2 = 0;
cin >> s;
while (s <= 220) {
 s = s + P;
 K1 = K1 + 1;
}
while (s >= Q) {
 s = s - Q;
 K2 = K2 + 1;
}
K1 = K1 + s; K2 = K2 + s;
cout << K1 << endl << K2;
return 0;
}

```

- 228) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. Найдите сумму натуральных значений переменной  $s$ , при которых программа выведет сначала 8, а потом 78?

```

#include <iostream>
using namespace std;
int main(){
 int P = 18, Q = 4, s;
 int K1 = 0, K2 = 0;
 cin >> s;
 while (s <= 300) {
 s = s + P;
 K1 = K1 + 1;
 }
 while (s >= Q) {
 s = s - Q;
 K2 = K2 + 1;
 }
 K1 = K1 + s; K2 = K2 + s;
 cout << K1 << endl << K2;
 return 0;
}

```

- 229) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наименьшем натуральном значении переменной  $s$  программа выведет сначала 36, а потом 335?

```

#include <iostream>
using namespace std;
int main(){
 int P = 13, Q = 4, s;
 int K1 = 0, K2 = 0;
 cin >> s;
 while (s != 1280) {
 s = s + P;
 K1 = K1 + 1;
 }
 while (s != Q + K1) {
 s = s - Q;
 K2 = K2 + 1;
 }
 K1 = K1 + s; K2 = K2 + s;
}

```

```

 cout << K1 << endl << K2;
 return 0;
}

```

- 230) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наименьшем натуральном значении переменной  $s$  программа выведет сначала 19, а потом 204?

```

#include <iostream>
using namespace std;
int main(){
 int P = 23, Q = 7, s;
 int K1 = 0, K2 = 0;
 cin >> s;
 while (s != 1350) {
 s = s + P;
 K1 = K1 + 1;
 }
 while (s != Q + K1) {
 s = s - Q;
 K2 = K2 + 1;
 }
 K1 = K1 + s; K2 = K2 + s;
 cout << K1 << endl << K2;
 return 0;
}

```

- 231) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наименьшем натуральном значении переменной  $s$  программа выведет сначала 314, а потом 470?

```

#include <iostream>
using namespace std;
int main(){
 int P = 29, Q = 11, s;
 int K1 = 0, K2 = 0;
 cin >> s;
 while (s != 2520) {
 s = s + P;
 K1 = K1 + 1;
 }
 while (s != Q + K1 + K2) {
 s = s - Q;
 K2 = K2 + 1;
 }
 K1 = K1 + s; K2 = K2 + s;
 cout << K1 << endl << K2;
 return 0;
}

```

- 232) (Е. Джобс) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. Сколько существует целых значений  $s$ , при которых после выполнения программы на экран будет выведено два числа 13, а затем 17?

```

#include <iostream>
using namespace std;
int main(){
 int P = 12, Q = 8, s, D=0;
 int K1 = 0, K2 = 0;
 cin >> s;

```

```

s = s + 21;
while (s >= 43) {
 s = s - 43;
 D += 1;
}
while (s < 0) {
 s = s + 43;
 D -= 1;
}
while (D <= 100) {
 D = D + P;
 K1 = K1 + 1;
}
while (D >= Q) {
 D = D - Q;
 K2 = K2 + 1;
}
K1 = K1 + D;
K2 = K2 + D;
cout << K1 << endl << K2;
}

```

- 233) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 15120. Укажите наибольшее возможное значение числа, выведенного вторым.

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 1; b = 0;
 while (x > 0) {
 d = x % 10;
 a *= d;
 if (d > 5)
 b += d;
 x /= 10;
 }
 cout << a << ' ' << b << endl;
}

```

- 234) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 5145. Укажите наибольшее возможное значение числа, выведенного вторым.

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 1; b = 0;
 while (x > 0) {
 d = x % 10;

```



```

 a *= d;
 if (d > 4)
 b += d;
 x /= 10;
}
cout << a << ' ' << b << endl;
}

```

- 235) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 529 200. Укажите наибольшее возможное значение числа, выведенного вторым.

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 1; b = 0;
 while (x > 0) {
 d = x % 10;
 a *= d;
 if (d > 3)
 b += d;
 x /= 10;
 }
 cout << a << ' ' << b << endl;
}

```

- 236) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 84 000 000. Укажите наибольшее возможное значение числа, выведенного вторым.

```

#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 1; b = 0;
 while (x > 0) {
 d = x % 10;
 a *= d;
 if (d > 2)
 b += d;
 x /= 10;
 }
 cout << a << ' ' << b << endl;
}

```

- 237) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 1 360 800 000. Укажите наибольшее возможное значение числа, выведенного вторым.

```

#include <iostream>
using namespace std;

```

```
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 1; b = 0;
 while (x > 0) {
 d = x % 10;
 a *= d;
 if (d > 1)
 b += d;
 x /= 10;
 }
 cout << a << ' ' << b << endl;
}
```

- 238) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 15120. Укажите наибольшее возможное значение числа, выведенного вторым.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 1; b = 0;
 while (x > 0) {
 d = x % 10;
 a *= d;
 if (d > 2)
 b += d;
 x /= 10;
 }
 cout << a << ' ' << b << endl;
}
```

- 239) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 5145. Укажите наибольшее возможное значение числа, выведенного вторым.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 1; b = 0;
 while (x > 0) {
 d = x % 10;
 a *= d;
 if (d > 1)
 b += d;
 x /= 10;
 }
 cout << a << ' ' << b << endl;
}
```

- 240) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 529 200. Укажите наибольшее возможное значение числа, выведенного вторым.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 1; b = 0;
 while (x > 0) {
 d = x % 10;
 a *= d;
 if (d > 4)
 b += d;
 x /= 10;
 }
 cout << a << ' ' << b << endl;
}
```

- 241) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 84 000 000. Укажите наибольшее возможное значение числа, выведенного вторым.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 1; b = 0;
 while (x > 0) {
 d = x % 10;
 a *= d;
 if (d > 3)
 b += d;
 x /= 10;
 }
 cout << a << ' ' << b << endl;
}
```

- 242) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 1 360 800 000. Укажите наибольшее возможное значение числа, выведенного вторым.

```
#include <iostream>
using namespace std;
int main()
{
 int x, a, b, d;
 cin >> x;
 a = 1; b = 0;
 while (x > 0) {
 d = x % 10;
```

```

 a *= d;
 if (d > 4)
 b += d;
 x /= 10;
}
cout << a << ' ' << b << endl;
}

```

- 243) (Е. Джобс) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает одно число. Найдите минимальное значение  $s$ , при котором после выполнения программы на экран будет выведено 58.

```

#include <iostream>
using namespace std;
int main(){
 long long s, k1=0, k2=0;
 cin >> s;
 while (s > 0) {
 int d = s % 10;
 if(d > 5){
 s = s - 1;
 k1 = k1 + 1;
 }
 else{
 s = s / 10;
 k2 = k2 + 1;
 }
 }
 cout << k1 + k2;
}

```

- 244) (ЕГЭ-2022) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа. Найдите наибольшее значение  $x$ , при котором программы напечатает сначала число 6, а потом снова число 6.

```

#include <iostream>
using namespace std;
int main()
{
 int x, N, S;
 cin >> x;
 N = 0;
 S = 0;
 while (x > 0) {
 N = N + 1;
 S = S + x % 2;
 if (N % 2 > 0)
 S = S + x % 4;
 x = x / 2;
 }
 cout << N << endl << S;
 return 0;
}

```

- 245) (ЕГЭ-2022) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа. Найдите наибольшее значение  $x$ , при котором программы напечатает сначала число 8, а потом снова число 21.

```

#include <iostream>
using namespace std;

```

```

int main() {
 int x, K1, K2, Q, P, L, M;
 cin >> x;
 Q = 6; P = 10;
 K1 = 0; K2 = 0;
 while (x <= 100) {
 K1 = K1 + 1;
 x = x + P;
 }
 while (x >= Q) {
 K2 = K2 + 1;
 x = x - Q;
 }
 L = x + K1;
 M = x + K2;
 cout << L << endl << M;
 return 0;
}

```

- 246) (А. Богданов) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа. Найдите наименьшее число  $x$ , при вводе которого алгоритм печатает числа 9 и 256.

```

#include <iostream>
using namespace std;
int main() {
 int x, a = 0, b = 0;
 cin >> x;
 while(x > 1) {
 a = a + 1;
 if(x%2 == 0)
 x = x / 2;
 else
 x = 3*x + 1;
 if (x > b) b = x;
 }
 cout <<a<<" "<<b;
}

```

- 247) (Е. Джобс) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. Сколько существует значений  $s$ , для которых программы выведет сначала 12, а потом 18?

```

#include <iostream>
using namespace std;
int main(){
 int P = 20, Q = 13, s;
 int K1 = 0, K2 = 0;
 cin >> s;
 while(s < 230) {
 s = s + P;
 K1 = K1 + 1;
 }
 while(s >= Q) {
 s = s - Q;
 K2 = K2 + 1;
 }
 cout << K1 << endl << K2;
}

```

```
 return 0;
}
```

- 248) (Е. Джобс) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает одно число. Укажите наибольшее число, не превосходящее 1000, после обработки которого на экране будет выведено число 70.

```
#include <iostream>
using namespace std;
int main() {
 int x, K, D;
 cin >> x;
 K = 9*x - 57;
 D = 9*x + 13;
 while (K*D > 0)
 if (K > D) K = K % D;
 else D = D % K;
 cout << K + D << endl;
}
```

## 22 (повышенный уровень, время – 7 мин)

**Тема:** Анализ программы, содержащей подпрограммы, циклы и ветвления.

**Что проверяется:**

Умение анализировать алгоритм, содержащий ветвление и цикл

1.6.1. Формализация понятия алгоритма.

1.1.4. Умение читать и отлаживать программы на языке программирования.

**Что нужно знать:**

- операции целочисленного деления ( $//$ ) и взятия остатка ( $\%$ )
- как работают операторы присваивания, циклы и условные операторы в языке программирования

### Пример задания:

**Р-13 (демо-2021).** Получив на вход число  $x$ , этот алгоритм печатает два числа:  $L$  и  $M$ . Укажите наибольшее число  $x$ , при вводе которого алгоритм печатает сначала 4, а потом 5.

```
x = int(input())
Q = 9
L = 0
while x >= Q:
 L = L + 1
 x = x - Q
M = x
if M < L:
 M = L
 L = x
print(L)
print(M)
```

**Решение (анализ):**

- 1) рассмотрим первый цикл:

```
Q = 9
L = 0
while x >= Q:
 L = L + 1
 x = x - Q
M = x
```

- 2) поскольку переменная  $L$  сначала равна 0 и увеличивается на 1 с каждым шагом цикла, она играет роль счётчика повторения цикла
- 3) на каждой итерации цикла мы вычитаем  $Q$  из  $x$  до тех пор, пока  $x$  не станет меньше  $Q$ ; фактически мы определяем, сколько раз «поместится»  $Q$  в  $x$
- 4) из предыдущих рассуждений следует, что это операция деления, при этом после завершения цикла в переменной  $L$  находится частное, а в  $x$  – остаток от деления введённого значения на  $Q$
- 5) рассмотрим строки после цикла:

```
if M < L:
 M = L
 L = x
print(L)
print(M)
```

- их роль состоит в том (это легко проверить ручной прокруткой), что значения  $M$  и  $L$  меняются местами, если только  $M < L$ ;
- 6) это означает, что значения частного и остатка (сначала  $L$ , потом  $M$ ) будут выведены в порядке возрастания
  - 7) нам нужно определить наибольшее число, при котором частное и остаток равны 4 и 5; для получения именно большего числа нам нужно взять как частное наибольшее из двух заданных чисел то есть 5 (соответственно, за остаток принять 4);
  - 8) поскольку делили на 9, искомое число равно  $5 \cdot 9 + 4 = 49$
  - 9) Ответ: **49**.
  - 10) понятно, что для получения минимального подходящего числа достаточно взять частное 4 и остаток 5, так что  $4 \cdot 9 + 5 = 41$ .

#### Решение (программа):

- 1) если анализ алгоритма провести не удалось, можно попробовать найти подходящее значение перебором (методом «грубой силы», *brute force*)
- 2) для этого выполним следующие действия:

- а) удаляем строчку, где вводится  $x$ :

```
x = int(input())
```

- б) вводим новую переменную  $x0$  и заключаем основную часть программы в цикл

```
x0 = 1
```

```
while True:
```

```
 x = x0
```

```
 # здесь нужно поместить запись основного алгоритма
```

```
 x0 += 1
```

- в) можно использовать и цикл по переменной, если вы уверены, что максимальное значение, скажем, не больше 1000:

```
for x0 in range(1, 1001):
```

```
 # здесь нужно поместить запись основного алгоритма
```

- г) заменяем строчки, в которых выводятся результаты, на проверку нужного нам случая; если должны быть выведены числа 4 и 5, выводим на экран число  $x0$ :

```
print(L)
```

```
print(M)
```

```
if L == 4 and M == 5:
```

```
 print(x0)
```

- 3) приведём полностью изменённую программу (добавленные строки выделены фоном):

```
for x0 in range(1, 1001):
```

```
 x = int(input())
```

```
 x = x0
```

```
 Q = 9
```

```
 L = 0
```

```
 while x >= Q:
```

```
 L = L + 1
```

```
 x = x - Q
```

```
 M = x
```

```
 if M < L:
```

```
 M = L
```

```
 L = x
```

```
 print(L)
```

```
 print(M)
```

```
 if L == 4 and M == 5:
```



```
print(x0)
```

- 4) при запуске программа выдаёт два числа – 41 и 49.
- 5) Ответ: 49.

#### Решение (программа с функцией):

- 1) вычислительную часть алгоритма можно оформить в виде функции, которая возвращает значение **True**, если число соответствует условию, и **False**, если не соответствует:

```
def LM45(x):
 Q = 9
 L = 0
 while x >= Q:
 L = L + 1
 x = x - Q
 M = x
 if M < L:
 M = L
 L = x
 return L == 4 and M == 5

for x in range(1,1001):
 if LM45(x):
 print(x)
```

#### Ещё пример задания:

**Р-12.** Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 5, а потом 16.

```
x = int(input())
a = 0; b = 1
while x > 0:
 if x % 2 > 0:
 a = a + x % 12
 else:
 b = b * (x % 12)
 x = x // 12
print(a); print(b)
```

#### Решение:

- 1) операторы в теле цикла говорят о том, что число разбивается на отдельные цифры в двенадцатеричной системе ( $x \bmod 12$ ) и на каждом шаге цикла отбрасывается последняя цифра двенадцатеричной записи числа ( $x := x \operatorname{div} 12$ )
- 2) если число нечётное ( $x \bmod 2 > 0$ ), то последняя цифра двенадцатеричной записи добавляется к  $a$ , если чётное –  $b$  умножается на эту последнюю цифру
- 3) поскольку основание 12 чётное, чётность числа равносильна чётности его последней цифры в двенадцатеричной системе счисления
- 4) поэтому в переменной  $a$  окажется сумма всех нечётных цифр двенадцатеричной записи числа, а в переменной  $b$  – произведение всех чётных цифр этой же двенадцатеричной записи
- 5) поскольку  $a$  нечётно, это может быть сумма из одной цифры 5
- 6) поскольку  $b$  больше 12, это произведение может быть построено из двух или более чётных чисел;

- 7) чтобы число было наименьшим, предположим, что чётных цифр две; одну из них нужно взять минимально возможную, то есть,  $16 = 2 \cdot 8$  (пара  $4 \cdot 4$  даст большее число)
- 8) таким образом, двенадцатеричная запись числа состоит из цифр 2, 5 и 8
- 9) чтобы число было минимальным, расставим эти цифры в порядке возрастания и переведем число в десятичную систему:  $258_{12} = 356$ .
- 10) Ответ: **356**.

### Ещё пример задания:

**Р-11.** Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ .

Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 4, а потом 5.

```
x = int(input())
a = 0; b = 1
while x > 0:
 if x % 2 > 0:
 a = a + x % 9
 else:
 b = b * (x % 9)
 x = x // 9
print(a)
print(b)
```

### Решение (С.С. Поляков, Саратов):

- 1) Что делает предложенный алгоритм? Пока в числе  $x$ , переведенном в девятеричную систему счисления, есть хотя бы одна цифра, в цикле выполняются следующие действия: если число  $x$  нечетное ( $x \bmod 2 > 0$ ), то переменная «а» увеличивается на последнюю цифру девятеричного представления числа  $x$  ( $a := a + x \bmod 9$ ), иначе предыдущее значение «b» умножается на последнюю цифру девятеричного представления числа  $x$  ( $b := b * (x \bmod 9)$ ). После этого отбрасывается последняя цифра девятеричного представления числа  $x$  ( $x := x \div 9$ ) и, если цифры в числе  $x$  ещё остались, то всё повторяется.
- 2) Для систем счисления с нечётным основанием (3, 5, 7, 9, ...) справедливо утверждение: число, записанное в системе счисления с нечетным основанием чётно тогда и только тогда, когда сумма всех его цифр чётна (поэтому судить о чётности числа по чётности его последней цифры в системе счисления с нечётным основанием нельзя).
- 3) Чтобы разобраться в том, что делает алгоритм, рассмотрим его работу при  $X = 8303_{10}$ .
  - а) Переведем 8303 в девятеричную систему счисления:  $8303_{10} = 12345_9$ .
  - б) Находим сумму цифр:  $X = 12345_9 \rightarrow S = 1+2+3+4+5=15$ ; сумма нечётная  $\rightarrow$  число нечётное  $\rightarrow a=0+5=5$ ;
  - в) Следующий шаг:  $X = 1234_9 \rightarrow S = 1+2+3+4=10$ . Сумма чётная  $\rightarrow$  число чётное  $\rightarrow b=1*4=4$ ;
  - г)  $X = 123_9 \rightarrow S = 1+2+3=6$ . Сумма чётная  $\rightarrow$  число чётное  $\rightarrow b=1*4*3=12$ ;
  - д)  $X = 12_9 \rightarrow S = 1+2=3$ . Сумма нечётная  $\rightarrow$  число нечётное  $\rightarrow a=0+5+2=7$ ;
  - е)  $X = 1_9 \rightarrow$  Сумма нечётная  $\rightarrow$  число нечётное  $\rightarrow a=0+5+2+1=8$ ;
  - ж) Таким образом,  $a=8$ ,  $b=12$ . Будет напечатано сначала число 8, а потом 12.
- 4) Решим нашу задачу. Наименьшее число должно начинаться с 1. Это даст нечётную сумму. Поэтому  $a=0+1=1$ . Чтобы получилось 4, нужна еще «3». Но 13 не подходит. Так как  $1+3=4$ , то это будет четная сумма, и  $b$  станет равным  $b=1*4=4$ . Нам нужно  $b=5$ . Значит, 15 подойдет, так как  $1+5=6$  тоже четная сумма. Теперь можно добавить и «3». 153 даст нечетную сумму  $1+5+3=9$ , поэтому  $a=0+1+3=4$ . Получилось число  $153_9$ . Осталось перевести его в десятичную систему счисления:  $153_9 = 129_{10}$ .
- 5) Ответ: **129**.

**Ещё пример задания:**

**Р-10.** Ниже приведён алгоритм. Укажите наименьшее число  $x$ , при вводе которого алгоритм напечатает сначала 2, потом 5.

```
x = int(input())
a = 0
b = 1
while x > 0:
 if x % 2 > 0:
 a += 1
 else:
 b += (x % 7)
 x //= 7
print(a)
print(b)
```

**Решение:**

- 11) видим, что в конце программы на экран выводятся переменные **a** и **b**
- 12) переменная **a** в начале равна нулю, затем при выполнении условия  $x \% 2 > 0$  увеличивается на 1, то есть **a** – счётчик
- 13) переменная **b** в начале равна 1, затем при нарушении условия  $x \% 2 > 0$  к ней добавляется  $(x \% 7)$  – последняя цифра записи числа **x** в системе счисления с основанием 7
- 14) в цикле
 

```
while x > 0:
 ...
 x //= 7
```

 значение переменной **x** делится на 7, пока число не станет равно 0; это значит, что от его семеричной записи по очереди отсекаются цифры, начиная с последней
- 15) изменение переменных **a** и **b** выполняется в условном операторе
 

```
if x % 2 > 0:
 a += 1
else:
 b += (x % 7);
```

 то есть если после очередного отсечения получилось нечётное число, увеличивается счётчик **a**, а если получилось чётное – к значению переменной **b** добавляется последняя цифра семеричной записи числа
- 16) поскольку фактически идёт работа с семеричной системой счисления, будем искать цифры нужного числа **x** в семеричной системе, а потом переведём его в десятичную систему
- 17) значение **a** – это количество нечётных чисел, полученных в процессе отсечения, а **b = 1 +** сумма последних цифр семеричной записи чётных чисел, полученных в процессе отсечения
- 18) поскольку основание системы нечётное, чётность числа зависит от чётности цифр: если число нечётных цифр чётное (0, 2, 4, ...), то всё число чётное, а если в семеричной записи числа нечётное число нечётных цифр, то число нечётное
- 19) поскольку счётчик **a** увеличивается два раза им значение **b** увеличивается, по крайней мере, один раз, сначала будем искать минимальное число среди трёхзначных чисел (в семеричной системе)
- 20) нам нужно расставить чётные и нечётные цифры в семеричной записи числа так, чтобы из получаемых в процессе отсечения чисел два были нечётными, а одно – чётным

- 21) **маской чётности** числа назовём последовательность нулей и единиц, в которой ноль обозначает чётную цифру, а единица – нечётную; например, маска 101 обозначает последовательность «нечётная – чётная – нечётная»
- 22) при добавлении в конец семеричной записи числа новой нечётной цифры (1 в маске) чётность меняется, а при добавлении чётной (0 в маске) – нет, поэтому исходное число с маской 010 в ходе работы алгоритма как раз даст два нечётных числа (с масками 010 и 01) и одно чётное (с маской 0)
- 23) так как  $b=5$ , последняя цифра единственного чётного числа равна  $b-1=4$ , то есть первая цифра числа в семеричной системе – 4
- 24) продолжаем строить минимальное число по маске 010: вторая цифра должна быть нечётной, минимальная нечётная цифра – 1
- 25) третья цифра чётная, минимальное возможное значение – 0
- 26) таким образом, получаем число  $410_7 = 203_{10}$
- 27) Ответ: **203**.

### Ещё пример задания:

**P-09.** Ниже приведён алгоритм. Укажите наибольшее число  $x$ , при вводе которого алгоритм напечатает сначала 3, потом 6.

```
x = int(input())
Q = 15
L = 0
while x >= Q:
 L += 1
 x -= Q
M = x
if M < L:
 M = L
 L = x
print(L)
print(M)
```

#### Решение:

1. видим, что в конце программы на экран выводятся переменные **L** и **M**
2. узнаём алгоритм в первой части программы:

```
L = 0
while x >= Q:
 L += 1
 x -= Q
```

Этот алгоритм вычисляет (с помощью последовательных вычитаний) частное **L** и остаток **x** от деления исходного значения **x** на **Q**.

3. Узнаём второй алгоритм:

```
M = x
if M < L:
 M = L
 L = x
```

Сначала то в **M** просто записывается значение **x** – остаток от деления исходного значения **x** на **Q**. Если  $x < L$ , то остаток и частное меняются местами так, чтобы в переменной **L** оказалось меньшее из двух значений, а в **M** – большее.

- 4.. таким образом, условию задачи при  $Q=15$  соответствуют два числа –  $15*3+6=51$  и  $15*6+3=93$ ; наибольшее из них – 93.
5. Ответ: 93.

### Ещё пример задания:

**Р-08.** Получив на вход число  $x$ , этот алгоритм печатает число  $M$ . Известно, что  $x > 100$ . Укажите наименьшее такое (т.е. большее 100) число  $x$ , при вводе которого алгоритм печатает 26.

```
x = int(input())
L = x
M = 65
if L % 2 == 0: M = 52
while L != M: # *
 if L > M: # *
 L -= M # *
 else: # *
 M -= L # *
print(M)
```

#### Решение:

- 1) видим, что в последней строке выводится на экран переменная **M**
- 2) ключевой момент решения: нужно узнать в строках программы, отмеченных знаком \* в комментариях, АЛГОРИТМ ЕВКЛИДА для вычисления наибольшего общего делителя (НОД) чисел, записанный в переменные **M** и **L**
- 3) введённое значение **x** записывается в переменную **L** и участвует в поиске НОД
- 4) в переменную **M** до начала цикла записывается 65, но если было введено чётное ( $L \% 2 = 0$ ) значение **x** (оно же **L**), значение **M** заменяется на 52
- 5) сначала предположим, что замены не было, и в **M** осталось значение 65; поскольку по условию алгоритм печатает 26, тогда получается, что **НОД(x, 65) = 26**; этого явно не может быть, потому что 65 не делится на 26
- 6) делаем вывод, что введено чётное значение **x** и произошла замена **M** на 52
- 7) итак, нужно найти чётное число **x**, большее 100, такое, что **НОД(x, 52) = 26**
- 8) первое число, большее 100, которое делится на 26 – это 104, но оно не подходит, потому что делится ещё и на 52, так что **НОД(x, 52) = 52**
- 9) поэтому берём следующее число, которое делится на 26:  $104 + 26 = 130$
- 10) Ответ: 130.

### Ещё пример задания:

**Р-07.** Ниже записан алгоритм. Укажите минимальное число  $x$ , при вводе которого алгоритм печатает 26391.

```
x = int(input())
K = 1
A = 0
B = 0
while x > 0 :
 if (x % 10) % 2 == 0 :
 A = A*10+x % 10
 else:
 K = K*10
```

```

 B = B*10 + x % 10
 x = x // 10
 A = A*K + B
 print(A)

```

**Решение:**

28) видим, что в последней строке выводится на экран переменная **A**, которая вычисляется в предыдущей строке по формуле **A = A\*K+B**

29) определим, сколько раз выполняется цикл **while** условие его продолжения –  $x > 0$ , с переменной **x** выполняется единственная операция – деление на 10 нацело:

```

while x>0 :
 ...
 x = x // 10

```

отсюда делаем вывод, что цикл выполняется столько раз, сколько цифр в десятичной записи введённого числа **x**

30) теперь посмотрим, что происходит внутри цикла: выбор варианта действия зависит от выполнения условия

```
(x % 10) % 2 == 0
```

здесь  $x \% 10$  – это последняя цифра  $x$ , в этом условии проверяется её чётность (делимость на 2)

31) итак, если последняя цифра числа чётная, выполняется оператор

```
A = A*10+x % 10
```

то есть, предыдущее значение **A** умножается на 10 и к результату добавляется последняя цифра **x** таким образом переменная **A** составляется из чётных цифр числа **x**, причём в обратном порядке, потому что новая цифра добавляется в конец числа, а предыдущие (которые были ближе к концу в записи числа **x**) продвигаются влево, в старшие разряды

32) теперь смотрим, как строится **B**: здесь всё то же самое, только нечётные цифры собираются в обратном порядке например, если исходное число было 12345, после окончания цикла мы получим **A=42** и **B=531**

33) но есть ещё переменная **K**, её начальное значение – 1, и с каждой найденной нечётной цифрой она умножается на 10, то есть **K=10** в степени, равной количеству нечётных цифр! для числа 12345 получим **K=1000**

34) в предпоследней строке по формуле **A = A\*K+B** собирается итоговое значение **A** для нашего примера (12345) мы получим **A = 42\*1000+531=42531**, то есть **K** служит для того, чтобы сдвинуть комбинацию чётных цифр в начало числа

35) итак, нам задано число 26391, поэтому в искомом числе есть чётные цифры (по порядку, слева направо) {6, 2} и нечётные цифры {1, 9, 3} (тоже по порядку)

36) как же расположить эти цифры, чтобы получилось минимальное число? для этого сравниваем первые числа в списках чётных и нечётных чисел, и записываем в ответ меньшее из них эту операцию повторяем, пока числа в обоих списках не кончатся помним, что менять порядок чётных и нечётных чисел нельзя!

37) в данном случае получается {1, 6, 2, 9, 3} = 16293.

38) Ответ: **16293**.

**Ещё пример задания:**

**Р-06.** Ниже записан алгоритм. Укажите наименьшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 4, а потом 2.

```

a = 0
b = 0

```

```
x = int(input())
while x > 0 :
 y = x % 10
 if y > 3 :
 a = a + 1
 if y < 8 :
 b = b + 1
 x = x // 10
print(a)
print(b)
```

**Решение:**

- 1) видим, что в последней строке выводятся на экран переменные **a** и **b**, поэтому сначала нужно определить, что они обозначают в программе
- 2) перед началом цикла переменные **a** и **b** обнуляются
- 3) на каждом шаге цикла при выполнении некоторых условий переменные **a** и **b** увеличиваются на 1, то есть представляют собой счётчики
- 4) увеличение переменных зависит от значения  $y = x \% 10$ , то есть от последней цифры числа
- 5) если последняя цифра числа больше 3, увеличивается счётчик **a**, если меньше 8 – счётчик **b**
- 6) в конце каждого шага цикла операция  $x = x // 10$  отсекает последнюю цифру в десятичной записи числа
- 7) цикл заканчивается, когда перестаёт выполняться условие  $x > 0$ , то есть, когда все цифры исходного числа отброшены
- 8) таким образом, делаем вывод: после завершения цикла в переменной **a** находится количество цифр, больших 3, в десятичной записи числа, а в переменной **b** – количество цифр, меньших 8
- 9) если было выведено 4 и 2, то в числе 4 цифры больше 3 и 2 цифры меньше 8
- 10) так как число пятизначное, есть  $4 + 2 = 5$  – одна цифра, которая больше 3 и меньше 8 одновременно она должна быть минимальной, поэтому эта цифра **4**
- 11) для того чтобы число было минимальным, ещё одна цифра должна быть минимальной и меньшей 3 – это старшая **1**, и три цифры минимальные из цифр, больших или равных 8, то есть **три цифры 8**
- 12) ответ: **14888**.

**Ещё пример задания:**

**Р-05.** Ниже записан алгоритм. Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 12?

```
x = int(input())
a = 0
b = 0
while x > 0 :
 a = a + 1
 b = b + (x % 10)
 x = x // 10
print(a)
print(b)
```

**Решение:**

- 1) видим, что в последней строке выводятся на экран переменные **a** и **b**, поэтому сначала нужно определить, что они обозначают в программе
- 2) перед началом цикла переменные **a** и **b** обнуляются
- 3) на каждом шаге цикла при выполнении некоторого условия переменная **a** увеличивается на 1, а **b** увеличивается на  $x \% 10$ , то есть, на остаток от деления **x** на 10 – это последняя цифра десятичной записи числа **x**
- 4) в конце каждого шага цикла операция  $x = x // 10$  отсекает последнюю цифру в десятичной записи числа
- 5) цикл заканчивается, когда перестаёт выполняться условие  $x > 0$ , то есть, когда все цифры исходного числа отброшены
- 6) таким образом, делаем вывод: после завершения цикла в переменной **a** находится количество цифр в десятичной записи числа, а в переменной **b** – их сумма
- 7) если было выведено 2 и 12, то в числе 2 цифры, и их сумма равна 12 таким образом, нам нужно найти все двузначные числа, в котором сумма значений цифр равна 12
- 8) число 12 может быть разложено на два слагаемых, меньших 10, как  
 $12 = 3 + 9 = 4 + 8 = 5 + 7 = 6 + 6 = 7 + 5 = 8 + 4 = 9 + 3$ ,  
нам подходят числа 39, 48, 57, 66, 75, 84 и 93
- 9) всего таких чисел - 7
- 10) ответ: **7**.

### Ещё пример задания:

**P-04.** Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 15.

```
x = int(input())
a = 0
b = 1
while x > 0 :
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print(a)
print(b)
```

#### Решение:

- 1) видим, что в последней строке выводятся на экран переменные **a** и **b**, поэтому сначала нужно определить, что они обозначают в программе
- 2) перед началом цикла переменная **a** обнуляется, а переменная **b** равна 1
- 3) на каждом шаге цикла при выполнении некоторого условия переменная **a** увеличивается на 1, а **b** умножается на  $x \% 10$ , то есть, на остаток от деления **x** на 10 – это последняя цифра десятичной записи числа **x**
- 4) в конце каждого шага цикла операция  $x = x // 10$  отсекает последнюю цифру в десятичной записи числа
- 5) цикл заканчивается, когда перестаёт выполняться условие  $x > 0$ , то есть, когда все цифры исходного числа отброшены
- 6) таким образом, делаем вывод: после завершения цикла в переменной **a** находится количество цифр в десятичной записи числа, а в переменной **b** – их произведение



- 7) если было выведено 2 и 15, то в числа 2 цифры, и их произведение равно 15 таким образом, нам нужно найти минимальное двузначное число, в котором произведение значений цифр равно 15
- 8) поскольку число 15 может быть разложено на два сомножителя, меньших 10, только как 3-5, минимальное подходящее число – 35.
- 9) ответ: **35**.

### Ещё пример задания:

**P-03.** Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 2.

```
x = int(input())
a = 0
b = 0
while x > 0 :
 c = x % 2
 if c == 0 :
 a = a + 1
 else:
 b = b + 1
 x = x // 10
print(a)
print(b)
```

#### Решение:

- 1) видим, что в последних строках выводятся на экран переменные  $a$  и  $b$ , поэтому сначала нужно определить, что они обозначают в программе
- 2) перед началом цикла обе переменные обнуляются
- 3) на каждом шаге цикла при выполнении некоторого условия переменная  $a$  увеличивается на 1, а если это условие не выполняется, то на 1 увеличивается  $b$  таким образом, обе переменных – счётчики
- 4) теперь посмотрим на условие  $c = 0$ : в предыдущей строке в переменную  $c$  записывается остаток от деления числа  $x$  на 2, то есть, переменная  $c$  определяет чётность числа или, что равносильно, **чётность его последней цифры**
- 5) если последняя цифра чётная, то увеличивается счётчик  $a$ , а если нечётная – увеличивается счётчик  $b$
- 6) в конце каждого шага цикла операция  $x = x // 10$  отсекает последнюю цифру в десятичной записи числа
- 7) таким образом, делаем вывод: после завершения цикла в переменной  $a$  находится количество чётных цифр в десятичной записи числа, а в переменной  $b$  – количество нечётных цифр
- 8) если было выведено 3 и 2, то в числа 5 цифр, из них 3 чётных и 2 нечётных таким образом, нам нужно найти минимальное пятизначное число, в котором 3 чётные и 2 нечётные цифры
- 9) минимальная чётная цифра – это 0, минимальная нечётная – 1 0 не может стоять на первом месте, поэтому число начинается с 1
- 10) для получения минимального числа после 1 должны идти нули и последняя цифра – снова 1
- 11) ответ: **10001**

**Ещё пример задания:**

**Р-02.** Ниже записан алгоритм. После выполнения алгоритма было напечатано 3 числа. Первые два напечатанных числа – это числа 9 и 81. Какое наибольшее число может быть напечатано третьим?

```
x, y = map(int, input().split())
if y > x :
 z = x
 x = y
 y = z
a = x
b = y
while b > 0 :
 r = a % b
 a = b
 b = r
print(a)
print(x)
print(y)
```

**Решение:**

- 1) сложность этой задачи состоит в том, чтобы разобраться в алгоритме
- 2) сначала вводятся два числа и переставляются так, чтобы в переменной **x** было наибольшее число, а в переменной **y** – наименьшее из двух:

```
if y > x :
 z = x
 x = y
 y = z
```

- 3) затем исходные значения копируются в переменные **a** и **b** и с ними выполняется следующий алгоритм

```
while b > 0 :
 r = a % b
 a = b
 b = r
```

его суть сводится к тому, что меньшее из двух чисел, **a** и **b**, каждый раз заменяется на остаток от деления большего на меньшее до тех пор, пока этот остаток не станет равен нулю

- 4) делаем вывод, что это классический [Алгоритм Евклида](#), который служит для вычисления наибольшего общего делителя (НОД) двух чисел это делитель в результате оказывается в переменной **a**
- 5) смотрим, что выводится на экран: сначала значение переменной **a** (наибольший общий делитель исходных чисел, НОД(**x**,**y**)), затем значение **x** (большее из исходных чисел) и значение **y** (меньшее из исходных чисел)
- 6) по условию первое число – 9, второе – 81, поэтому третье число должно быть меньше, чем 81, и НОД(81,**y**) = 9
- 7) наибольшее число, которое меньше 81 и делится на 9, равно 72 (обратите внимание, что исходные числа не могут быть равны, потому что в этом случае их НОД был бы равен 81)
- 8) ответ: **72**

**Ещё пример задания:**

**Р-01.** Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0
M = 0
while x > 0 :
 L = L+1
 if M < (x % 10) :
 M = x % 10
 x = x // 10
print(L)
print(M)
```

**Решение:**

- 1) для решения задачи необходимо понять, что делает эта программа
- 2) если это не видно сразу, можно выполнить ручную прокрутку для какого-то простого числа, например, для числа 251:

| оператор             | условие            | x   | L | M |
|----------------------|--------------------|-----|---|---|
| x = int(input())     |                    | 251 | ? | ? |
| L = 0 M = 0          |                    |     | 0 | 0 |
| while x > 0 :...     | 251 > 0? да        |     |   |   |
| L = L+1              |                    |     | 1 |   |
| if M < (x % 10) :... | M < (251 % 10)? да |     |   |   |
| M = x % 10           |                    |     |   | 1 |
| x = x // 10          |                    | 25  |   |   |
| while x > 0 :...     | 25 > 0? да         |     |   |   |
| L = L+1              |                    |     | 2 |   |
| if M < (x % 10) :... | M < (25 % 10)? да  |     |   |   |
| M = x % 10           |                    |     |   | 5 |
| x = x // 10          |                    | 2   |   |   |
| while x > 0 :...     | 2 > 0? да          |     |   |   |
| L = L+1              |                    |     | 3 |   |
| if M < (x % 10) :... | M < (2 % 10)? нет  |     |   |   |
| x = x // 10          |                    | 0   |   |   |
| while x > 0 :...     | 0 > 0? нет         |     |   |   |
| print(L) print(M)    |                    |     | 3 | 5 |

- 3) можно догадаться, что в результате работы программы в переменной L окажется число цифр числа, а в переменной M – наибольшая цифра, но это предположение нужно постараться доказать
- 4) нужно вспомнить (и запомнить), что для целого числа  $x$  остаток от деления на 10 ( $x \% 10$ ) – это последняя цифра в десятичной записи числа, а целочисленное деление ( $x // 10$ ) отсекает последнюю цифру, то есть из 123 получается 12
- 5) рассмотрим цикл, число шагов которого зависит от изменения переменной  $x$ :

```
while x > 0 :
 ...
 x = x // 10 { отсечение последней цифры }
```

здесь оставлены только те операторы, которые влияют на значение  $x$

- 6) из приведенного цикла видно, что на каждом шаге от десятичной записи **х** отсекается последняя цифра до тех пор, пока все цифры не будут отсечены, то есть **х** не станет равно 0 поэтому **цикл выполняется столько раз, сколько цифр в десятичной записи введенного числа**

- 7) на каждом шаге цикла переменная **L** увеличивается на 1:

```
L = L+1
```

других операторов, меняющих значение **L**, в программе нет поэтому после завершения цикла **в переменной L действительно находится количество цифр**

- 8) теперь разберемся с переменной **M**, которая сначала равна 0 оператор, в котором она меняется, выглядит так:

```
if M < (x % 10) :
 M = x % 10
```

учитывая, что **x % 10** – это последняя цифра десятичной записи числа, получается что если эта цифра больше, чем значение **M**, она записывается в переменную **M**

- 9) этот оператор выполняется в цикле, причем выражение **x % 10** по очереди принимает значения всех цифр исходного числа поэтому после завершения цикла **в переменной M окажется наибольшая из всех цифр**, то есть наша догадка подтверждается
- 10) итак, по условию задачи фактически требуется найти наибольшее трехзначное число, в котором наибольшая цифра – 7 очевидно, что это 777.
- 11) ответ: **777**.

#### Возможные ловушки и проблемы:

- это очень неплохая задача на понимание, тут достаточно сложно «вызубрить» метод решения, можно только освоить последовательность (системность) анализа
- ручной прокрутки в такой задаче недостаточно, по её результатам можно угадать алгоритм, но можно и не угадать в критическом случае можно сделать ручную прокрутку для нескольких чисел им попытаться понять закономерность

### Ещё пример задания:

**P-00.** Ниже записана программа. Получив на вход число *x*, эта программа печатает два числа, *L* и *M*. Укажите наибольшее из таких чисел *x*, при вводе которых алгоритм печатает сначала 3, а потом 120.

```
x = int(input())
L = 0
M = 1
while x > 0 :
 L = L+1
 M = M*(x % 8)
 x = x // 8
print(L)
print(M)
```

#### Решение:

- 1) для решения задачи необходимо понять, что делает эта программа повторяя рассуждения из предыдущего примера, выясняем, что
- переменная **L** с каждым шагом цикла увеличивается на 1
  - переменная **x** на каждом шаге цикла делится на 8 и остаток отбрасывается

поэтому можно сделать вывод, что в конце цикла переменная **L** будет равна **количеству цифр** введенного числа, записанного в *восьмеричной* системе счисления таким образом, восьмеричная запись числа содержит ровно 3 цифры

- 2) выражение  $x \% 8$  – это последняя цифра восьмеричной записи числа на каждом шаге цикла переменная **M** умножается на эту величину, поэтому в результате в **M** будет записано **произведение всех цифр** восьмеричной записи введенного числа
- 3) по условию это произведение равно 120, то есть  $120 = a \cdot b \cdot c$ , где  $a, b$  и  $c$  – числа от 0 до 7 (которые в восьмеричной системе счисления записываются одной цифрой)
- 4) поскольку нам нужно наибольшее число, перебираем делители числа 120, начиная со старшей цифры – 7 видим, что 120 на 7 не делится, поэтому такой цифры в восьмеричной записи числа нет
- 5) но 120 делится на 6, поэтому старшей цифрой может быть 6 – только в том случае, когда второй сомножитель можно представить в виде произведения двух чисел в интервале 1..6
- 6) делим 120 на 6, получаем 20 это число представляется как произведение 5 и 4, каждое из этих чисел записывается в виде одной восьмеричной цифры, то есть, они нам подходят
- 7) вспомним, что нас интересует максимальное число, поэтому цифры нужно выстроить в порядке убывания:  $654_8$
- 8) заметим, что мы получили число в восьмеричной системе, а ответ нужно дать в десятичной переводим:  $654_8 = 6 \cdot 8^2 + 5 \cdot 8^1 + 4 \cdot 8^0 = 428$ .
- 9) ответ: **428**.

**Возможные ловушки и проблемы:**

- поскольку в цикле идет деление на 8, мы получаем цифры числа в восьмеричной системе каждая из них должна быть в интервале 0..7 (не может быть 8 и 9)
- на последнем шаге нужно не забыть перевести число из восьмеричной системы в десятичную

## Задачи для тренировки<sup>1</sup>:

- 1) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 M = M + (x % 10)
 x = x // 10
print("%d\n%d" % (L, M))
```

- 2) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 0:
 M = M + (x % 10)
 x = x // 10
print("%d\n%d" % (L, M))
```

- 3) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 0.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 0:
 M = M + (x % 10)
 x = x // 10
print("%d\n%d" % (L, M))
```

- 4) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 1:
 M = M + (x % 10)
 x = x // 10
print("%d\n%d" % (L, M))
```

<sup>1</sup> Источники заданий:

1. Авторские разработки.
2. Тренировочные и диагностические работы МИОО.
3. Путимцева Ю.С. Информатика. Диагностические работы в формате ЕГЭ 2012. М.: МЦНМО, 2012.
4. Евич Л.Н., Кулабухов С.Ю. Информатика и ИКТ. Подготовка к ЕГЭ-2014. — Ростов-на-Дону: Легион, 2013.

- 5) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 0:
 M = M + (x % 10) // 2
 x = x // 10
print("%d\n%d" % (L, M))
```

- 6) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 1:
 M = M + (x % 10) // 2
 x = x // 10
print("%d\n%d" % (L, M))
```

- 7) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if M < x:
 M = x % 10
 x = x // 10
print("%d\n%d" % (L, M))
```

- 8) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if (M < x) and (x % 2 == 0):
 M = x % 10
 x = x // 10
print("%d\n%d" % (L, M))
```

- 9) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 10.

```
x = int(input())
L = 0; M = 0
while x > 0:
```

```

L = L + 1
if (M < x) and (x % 2 == 1):
 M = (x % 10) * 2
x = x // 10
print("%d\n%d" % (L, M))

```

- 10) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 10.

```

x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if (M < x):
 M = (x % 10) * 2
 x = x // 10
print("%d\n%d" % (L, M))

```

- 11) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 72.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 12) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 14.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 13) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 14) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 0.

```

x = int(input())

```



```
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))
```

- 15) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 M = M + (x % 10)
 x = x // 10
print("%d\n%d" % (L, M))
```

- 16) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 0:
 M = M + (x % 10)
 x = x // 10
print("%d\n%d" % (L, M))
```

- 17) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 0.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 0:
 M = M + (x % 10)
 x = x // 10
print("%d\n%d" % (L, M))
```

- 18) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 1:
 M = M + (x % 10)
 x = x // 10
print("%d\n%d" % (L, M))
```

- 19) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 0:
 M = M + (x % 10) // 2
 x = x // 10
print("%d\n%d" % (L, M))
```

- 20) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 1:
 M = M + (x % 10) // 2
 x = x // 10
print("%d\n%d" % (L, M))
```

- 21) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if M < x:
 M = x % 10
 x = x // 10
print("%d\n%d" % (L, M))
```

- 22) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if (M < x) and (x % 2 == 0):
 M = x % 10
 x = x // 10
print("%d\n%d" % (L, M))
```

- 23) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 10.

```
x = int(input())
L = 0; M = 0
while x > 0:
```

```

L = L + 1
if (M < x) and (x % 2 == 1):
 M = (x % 10) * 2
x = x // 10
print("%d\n%d" % (L, M))

```

- 24) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 28.

```

x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if (M < x):
 M = M + (x % 10) * 2
 x = x // 10
print("%d\n%d" % (L, M))

```

- 25) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 72.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 26) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 14.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 27) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 28) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 36.

```

x = int(input())

```

```
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))
```

- 29) (Д.Ю. Мельникова, г. Саратов) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 24.

```
x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 8)
 x = x // 8
print("%d\n%d" % (a, b))
```

- 30) (Д.Ю. Мельникова, г. Саратов) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 10.

```
x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 8)
 x = x // 8
print("%d\n%d" % (a, b))
```

- 31) (Д.Ю. Мельникова, г. Саратов) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 6.

```
x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 6)
 x = x // 6
print("%d\n%d" % (a, b))
```

- 32) (Д.Ю. Мельникова, г. Саратов) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 9.

```
x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 5)
 x = x // 5
print("%d\n%d" % (a, b))
```

- 33) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 21.

```
x = int(input())
```

```
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))
```

- 34) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 35.

```
x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))
```

- 35) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 4.

```
x = int(input())
L = 0; M = 9
while x > 5:
 L = L + 1
 if M > (x % 10):
 M = x % 10
 x = x // 10
print("%d\n%d" % (L, M))
```

- 36) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0; M = 0
while x > 5:
 L = L + 1
 if M < (x % 10):
 M = x % 10
 x = x // 10
print("%d\n%d" % (L, M))
```

- 37) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 6, а потом 5.

```
x = int(input())
a = 0; b = 0
while x > 0:
 a = a + 2
 b = b + (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))
```

- 38) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 8, а потом 19.

```
x = int(input())
a = 0; b = 0
while x > 0:
 a = a + 2
 b = b + (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))
```

- 39) Ниже записан алгоритм. После выполнения алгоритма было напечатано 3 числа. Первые два напечатанных числа – это числа 7 и 42. Какое наибольшее число может быть напечатано третьим?

```
x = int(input())
y = int(input())
if (y > x):
 z = x
 x = y
 y = z
a = x; b = y
while b > 0:
 r = a % b
 a = b
 b = r
print("%d\n%d\n%d" % (a, x, y))
```

- 40) Ниже записан алгоритм. После выполнения алгоритма было напечатано 3 числа. Первые два напечатанных числа – это числа 13 и 65. Какое наибольшее число может быть напечатано третьим?

```
x = int(input())
y = int(input())
if (y > x):
 z = x
 x = y
 y = z
a = x; b = y
while b > 0:
 r = a % b
 a = b
 b = r
print("%d\n%d\n%d" % (a, x, y))
```

- 41) Ниже записан алгоритм. Укажите наименьшее из таких чисел  $N$ , при вводе которых алгоритм напечатает 17.

```
N = int(input())
for i in range(1, N):
 if N % i == 0:
 q = i
print(q)
```

- 42) (<http://ege.yandex.ru>) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите, сколько есть таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 12.

```
x = int(input())
```

```
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))
```

- 43) (<http://ege.yandex.ru>) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 8.

```
x = int(input())
a = 0; b = 0
while x > 0:
 a = a + 1
 b = b + (x % 100)
 x = x // 100
print("%d\n%d" % (a, b))
```

- 44) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 8.

```
x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 100)
 x = x // 100
print("%d\n%d" % (a, b))
```

- 45) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 14, а потом 6.

```
x = int(input())
a = 0; b = 10
while x > 0:
 c = x % 10
 a = a + c
 if c < b:
 b = c
 x = x // 10
print("%d\n%d" % (a, b))
```

- 46) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 15, а потом 5.

```
x = int(input())
a = 0; b = 10
while x > 0:
 c = x % 10
 a = a + c
 if c < b:
 b = c
 x = x // 10
print("%d\n%d" % (a, b))
```

- 47) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ .  
Укажите наибольшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 13, а потом 3.

```
x = int(input())
a = 0; b = 10
while x > 0:
 c = x % 10
 a = a + c
 if c < b:
 b = c
 x = x // 10
print("%d\n%d" % (a, b))
```

- 48) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ .  
Укажите наименьшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 19, а потом 4.

```
x = int(input())
a = 0; b = 10
while x > 0:
 c = x % 10
 a = a + c
 if c < b:
 b = c
 x = x // 10
print("%d\n%d" % (a, b))
```

- 49) Ниже записан алгоритм. Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 2.

```
x = int(input())
a = 0; b = 0
while x > 0:
 c = x % 2
 if c == 0:
 a = a + 1
 else:
 b = b + 1
 x = x // 10
print("%d\n%d" % (a, b))
```

- 50) Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 0.

```
x = int(input())
a = 0; b = 0
while x > 0:
 c = x % 2
 if c == 0:
 a = a + 1
 else:
 b = b + 1
 x = x // 10
print("%d\n%d" % (a, b))
```

- 51) Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 2.

```
x = int(input())
a = 0; b = 0
```



```
while x > 0:
 c = x % 2
 if c == 0:
 a = a + 1
 else:
 b = b + 1
 x = x // 8
print("%d\n%d" % (a, b))
```

- 52) Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 0.

```
x = int(input())
a = 0; b = 0
while x > 0:
 c = x % 2
 if c == 0:
 a = a + 1
 else:
 b = b + 1
 x = x // 6
print("%d\n%d" % (a, b))
```

- 53) Получив на вход число  $x$ , этот алгоритм печатает два числа  $K$  и  $R$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 3.

```
x = int(input())
K = 0; R = 9
y = x % 10
while x > 0:
 K = K + 1
 if R > (x % 10):
 R = x % 10
 x = x // 10
R = y - R
print("%d\n%d" % (K, R))
```

- 54) Получив на вход число  $x$ , этот алгоритм печатает два числа  $K$  и  $R$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
K = 0; R = 9
y = x % 10
while x > 0:
 K = K + 1
 if R > (x % 10):
 R = x % 10
 x = x // 10
R = y - R
print("%d\n%d" % (K, R))
```

- 55) Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 13.

```
x = int(input())
a = 0; b = 0
while x > 0:
 a = a + 1
 b = b + (x % 100)
```

```
x = x // 100
print("%d\n%d" % (a, b))
```

- 56) Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 15.

```
x = int(input())
a = 0; b = 0
while x > 0:
 a = a + 1
 b = b + (x % 100)
 x = x // 100
print("%d\n%d" % (a, b))
```

- 57) Ниже записан алгоритм. Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 15?

```
x = int(input())
a = 0; b = 0
while x > 0:
 a = a + 1
 b = b + (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))
```

- 58) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает числа:  $a$  и  $b$ . Укажите наименьшее положительное пятизначное число  $x$ , при котором после выполнения алгоритма будет напечатано сначала 5, а потом 2.

```
x = int(input())
a = 0; b = 10
while x > 0:
 y = x % 10
 x = x // 10
 if (y > a):
 a = y
 if (y < b):
 b = y
print("%d\n%d" % (a, b))
```

- 59) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает числа:  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которого после выполнения алгоритма будет напечатано сначала 2, а потом 22.

```
x = int(input())
a = 0; b = 0
while x > 0:
 a = a + 1
 b = b + (x % 100)
 x = x // 100
print("%d\n%d" % (a, b))
```

- 60) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 5.

```
x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 100)
```

```
x = x // 100
print("%d\n%d" % (a, b))
```

- 61) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 18.

```
x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 100)
 x = x // 100
print("%d\n%d" % (a, b))
```

- 62) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наименьшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 5, а потом 3.

```
x = int(input())
a = 0; b = 0
while x > 0:
 y = x % 10
 if y > 3: a = a + 1
 if y < 8: b = b + 1
 x = x // 10
print("%d\n%d" % (a, b))
```

- 63) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наибольшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 4, а потом 3.

```
x = int(input())
a = 0; b = 0
while x > 0:
 y = x % 10
 if y > 3: a = a + 1
 if y < 8: b = b + 1
 x = x // 10
print("%d\n%d" % (a, b))
```

- 64) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наибольшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 5, а потом 4.

```
x = int(input())
a = 0; b = 0
while x > 0:
 y = x % 10
 if y > 4: a = a + 1
 if y < 6: b = b + 1
 x = x // 10
print("%d\n%d" % (a, b))
```

- 65) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наименьшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 3, а потом 4.

```
x = int(input())
a = 0; b = 0
while x > 0:
 y = x % 10
 if y > 4: a = a + 1
 if y < 6: b = b + 1
 x = x // 10
print("%d\n%d" % (a, b))
```

- 66) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наименьшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 2, а потом 4.

```
x = int(input())
a = 0; b = 0
while x > 0:
 y = x % 10
 if y > 3: a = a + 1
 if y < 7: b = b + 1
 x = x // 10
print("%d\n%d" % (a, b))
```

- 67) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наибольшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 2, а потом 4.

```
x = int(input())
a = 0; b = 0
while x > 0:
 y = x % 10
 if y > 4: a = a + 1
 if y < 7: b = b + 1
 x = x // 10
print("%d\n%d" % (a, b))
```

- 68) Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 6.

```
x = int(input())
L = 0; M = 0
while x > 0:
 M = M + 1
 if x % 2 != 0:
 L = L + 1
 x = x // 2
print("%d\n%d" % (L, M))
```

- 69) Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 11.

```
x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 1000)
 x = x // 1000
print("%d\n%d" % (a, b))
```

- 70) Ниже записан алгоритм. Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 13.

```
x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 2
 b = b * (x % 1000)
 x = x // 1000
print("%d\n%d" % (a, b))
```

- 71) Ниже записан алгоритм. Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 12?

```
x = int(input())
a = 0; b = 0
while x > 0:
 a = a + 1
 b = b + (x % 100)
 x = x // 100
print("%d\n%d" % (a, b))
```

- 72) Получив на вход число  $x$ , этот алгоритм печатает два числа:  $L$  и  $M$ . Укажите наибольшее число  $x$ , при вводе которого алгоритм печатает сначала 5, а потом 8.

```
x = int(input())
L = 0
M = 0
while x > 0:
 M = M + 1
 if x % 2 != 0:
 L = L + 1
 x = x // 2
print(L)
print(M)
```

- 73) Получив на вход число  $x$ , этот алгоритм печатает число  $M$ . Известно, что  $x > 100$ . Укажите наименьшее такое (т. е. большее 100) число  $x$ , при вводе которого алгоритм печатает 15.

```
x = int(input())
L = x - 30
M = x + 30
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)
```

- 74) Получив на вход число  $x$ , этот алгоритм печатает число  $M$ . Известно, что  $x > 100$ . Укажите наименьшее такое (т. е. большее 100) число  $x$ , при вводе которого алгоритм печатает 30.

```
x = int(input())
L = x - 30
M = x + 30
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)
```

- 75) Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 18.

```
x = int(input())
a = 0
b = 0
while x > 0:
 a = a + 1
 if x % 2 == 0:
```

```
 b = b + x % 10
 x = x // 10
 print(a)
 print(b)
```

- 76) Получив на вход число **x**, этот алгоритм печатает два числа **a** и **b**. Укажите наименьшее из таких чисел **x**, при вводе которых алгоритм печатает сначала 45, а потом 5.

```
x = int(input())
a = 1
b = 10
while x > 0:
 c = x % 10
 a = a*c
 if c < b:
 b = c
 x = x // 10
print(a)
print(b)
```

- 77) Получив на вход число **x**, этот алгоритм печатает число **M**. Известно, что **x** > 100. Укажите наименьшее такое (т. е. большее 100) число **x**, при вводе которого алгоритм печатает 2.

```
x = int(input())
L = x - 12
M = x + 12
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)
```

- 78) Получив на вход число **x**, этот алгоритм печатает число **M**. Известно, что **x** > 100. Укажите наименьшее такое (т. е. большее 100) число **x**, при вводе которого алгоритм печатает 11.

```
x = int(input())
L = x - 21
M = x + 12
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)
```

- 79) Получив на вход число **x**, этот алгоритм печатает число **M**. Известно, что **x** > 100. Укажите наименьшее такое (т. е. большее 100) число **x**, при вводе которого алгоритм печатает 35.

```
x = int(input())
L = x - 15
M = x + 20
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
```

```
print(M)
```

- 80) Получив на вход число  $x$ , этот алгоритм печатает число  $M$ . Известно, что  $x > 100$ . Укажите наименьшее такое (т. е. большее 100) число  $x$ , при вводе которого алгоритм печатает 9.

```
x = int(input())
L = x - 18
M = x + 36
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)
```

- 81) Получив на вход число  $x$ , этот алгоритм печатает число  $M$ . Известно, что  $x > 100$ . Укажите наименьшее такое (т. е. большее 100) число  $x$ , при вводе которого алгоритм печатает 35.

```
x = int(input())
L = x - 20
M = x + 15
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)
```

- 82) Получив на вход число  $x$ , этот алгоритм печатает число  $M$ . Известно, что  $x > 100$ . Укажите наименьшее такое (т. е. большее 100) число  $x$ , при вводе которого алгоритм печатает 4.

```
x = int(input())
L = x - 16
M = x + 32
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)
```

- 83) Получив на вход число  $x$ , этот алгоритм печатает число  $M$ . Известно, что  $x > 100$ . Укажите наименьшее такое (т. е. большее 100) число  $x$ , при вводе которого алгоритм печатает 16.

```
x = int(input())
L = x - 16
M = x + 16
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)
```

- 84) (Д.Ф. Мұфаззалов) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 2.

```
x = int(input())
```

```
a = 0
b = 0
i = 0
while x > 0:
 i = i + 1
 c = x % 10
 if i % 2 == 0:
 a = a + c
 else:
 b = b + c
 x = x // 10
print(a)
print(b)
```

- 85) (Д.Ф. Мухаззалов) Ниже приведён алгоритм. Укажите наименьшее из таких чисел **x**, при вводе которых алгоритм печатает сначала 4, а потом 5.

```
x = int(input())
b = 0
i = 0
while x > 0:
 if i % 2 > 0:
 b = b + x % 10
 x = x // 10
 i = i + 1
print(i)
print(b)
```

- 86) (Д.Ф. Мухаззалов) Ниже приведён алгоритм. Укажите наименьшее из таких чисел **x**, при вводе которых алгоритм печатает сначала 4, а потом 5.

```
x = int(input())
b = 0
i = 0
while x > 0:
 b = b + x % 10
 x = x // 10
 i = i + 1
print(i)
print(b)
```

- 87) (Д.Ф. Мухаззалов) Ниже приведён алгоритм. Укажите наименьшее из таких чисел **x**, при вводе которых алгоритм печатает сначала 3, а потом 2.

```
x = int(input())
a = 0
b = 0
c = 0
while x > 0:
 i = i + 1
 if i % 2 == 0:
 a = a + c
 else:
 b = b + c
 c = x % 10
```



```

 x = x // 10
print(a)
print(b)

```

- 88) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 15.

```

x = int(input())
a = 0; b = 0; d = 0
while x > 0:
 if d % 2 == 0:
 a += x % 10
 else:
 b += x % 10
 x = x // 10
 d += 1
print(a, b)

```

- 89) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 14, а потом 12.

```

x = int(input())
a = 0; b = 0; d = 0
while x > 0:
 if d % 2 == 0:
 a += x % 10
 else:
 b += x % 10
 x = x // 10
 d += 1
print(a, b)

```

- 90) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , большее, чем 200, при вводе которого алгоритм напечатает 70.

```

x = int(input())
L = 2*x-30
M = 2*x+40
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)

```

- 91) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , большее, чем 200, при вводе которого алгоритм напечатает 50.

```

x = int(input())
L = 2*x-20
M = 2*x+30
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)

```

- 92) Ниже приведён алгоритм. Укажите наименьшее натуральное число  $x$ , при вводе которого алгоритм напечатает четырёхзначное число.

```
x = int(input())
x0 = x
N = 0
while x > 0:
 d = x % 2
 N = 10 * N + d
 x = x // 2
N += x0
print(N)
```

- 93) Ниже приведён алгоритм. Укажите наименьшее натуральное число  $x$ , при вводе которого алгоритм напечатает пятизначное число.

```
x = int(input())
x0 = x
N = 0
while x > 0:
 d = x % 2
 N = 10 * N + d
 x = x // 2
N += x0
print(N)
```

- 94) Ниже приведён алгоритм. Укажите наименьшее натуральное число  $x$ , при вводе которого алгоритм напечатает четырёхзначное число.

```
x = int(input())
x0 = x
N = 0
while x > 0:
 d = x % 3
 N = 10 * N + d
 x = x // 3
N += x0
print(N)
```

- 95) Ниже приведён алгоритм. Укажите наименьшее натуральное число  $x$ , при вводе которого алгоритм напечатает пятизначное число.

```
x = int(input())
x0 = x
N = 0
while x > 0:
 d = x % 3
 N = 10 * N + d
 x = x // 3
N += x0
print(N)
```

- 96) Ниже приведён алгоритм. Укажите наименьшее натуральное число  $x$ , при вводе которого алгоритм напечатает шестизначное число.

```
x = int(input())
x0 = x
N = 0
while x > 0:
 d = x % 3
```

```
N = 10*N + d
x = x // 3
N += x0
print(N)
```

- 97) (А.Н. Носкин) Ниже приведён алгоритм. Укажите наименьшее число  $x$ , при вводе которого алгоритм напечатает число, большее 10200.

```
x = int(input())
x0 = x
N = 0
while x > 0:
 d = x % 5
 N = 10*N + d
 x = x // 5
N += x0
print(N)
```

- 98) (А.Н. Носкин) Ниже приведён алгоритм. Укажите наименьшее число  $x$ , при вводе которого алгоритм напечатает число, большее, чем 10300.

```
x = int(input())
x0 = x
N = 0
while x > 0:
 d = x % 4
 N = 10*N + d
 x = x // 4
N += x0
print(N)
```

- 99) Ниже приведён алгоритм. Укажите наибольшее число  $x$ , при вводе которого алгоритм напечатает сначала 3, потом 5.

```
x = int(input())
Q = 6
L = 0
while x >= Q:
 L = L + 1
 x = x - Q
M = x
if M < L:
 M = L
 L = x
print(L)
print(M)
```

- 100) Ниже приведён алгоритм. Укажите наибольшее число  $x$ , при вводе которого алгоритм напечатает сначала 8, потом 11.

```
x = int(input())
Q = 16
L = 0
while x >= Q:
 L = L + 1
 x = x - Q
M = x
if M < L:
 M = L
```

```
L = x
print(L)
print(M)
```

- 101) (Д.В. Богданов) Укажите наименьшее натуральное число  $x$ , при вводе которого будет два раза напечатано число 3.

```
x = int(input())
a = 0
b = 1
while x > 0:
 if x % 3 > 0:
 a += 1
 if x % 3 > 1:
 b += 1
 x = x // 10
print(a)
print(b)
```

- 102) Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 24?

```
x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b*(x%10)
 x = x//10
print(a, "\n", b)
```

- 103) Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 0?

```
x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b*(x%10)
 x = x//10
print(a, "\n", b)
```

- 104) (Д.Ф. Муфаззалов) Укажите наибольшее двузначное натуральное число, при вводе которого приведенная ниже программа напечатает число 0.

```
n = int(input())
i = 0
while n > 0:
 i = i + n % 16
 n = n // 16
print(i % 15)
```

- 105) (Д.Ф. Муфаззалов) Укажите наименьшее трехзначное натуральное число, при вводе которого приведенная ниже программа напечатает число 0.

```
n = int(input())
i = 0
while n > 0:
 i = i + n % 20
 n = n // 20
print(i % 19)
```

- 106) (Д.Ф. Муфаззалов) Укажите количество двузначных натуральных чисел, при вводе которых приведенная ниже программа напечатает число 0.

```
n = int(input())
i = 0
while n > 0:
 i = i + n % 8
 n = n // 8
print(i % 7)
```

- 107) (Д.Ф. Муфаззалов) Укажите количество двузначных натуральных чисел, при вводе которых приведенная ниже программа напечатает число, отличное от 0.

```
n = int(input())
i = 0
while n > 0:
 i = i + n % 9
 n = n // 9
print(i % 8)
```

- 108) Укажите наименьшее натуральное число, при вводе которого эта программа напечатает сначала 1, потом 4.

```
x = int(input())
a=0; b=0
while x > 0:
 if x%2 == 0:
 a += 1
 else:
 b += x%6
 x = x//6
print(a, b)
```

- 109) Укажите наименьшее натуральное число, при вводе которого эта программа напечатает сначала 2, потом 6.

```
x = int(input())
a=0; b=0
while x > 0:
 if x%2 == 0:
 a += 1
 else:
 b += x%6
 x = x//6
print(a, b)
```

- 110) Укажите наименьшее натуральное число, при вводе которого эта программа напечатает сначала 1, потом 5.

```
x = int(input())
a=0; b=0
while x > 0:
 if x%2 == 0:
 a += 1
 else:
 b += x%4
 x = x//4
print(a, b)
```

- 111) Укажите наименьшее натуральное число, при вводе которого эта программа напечатает сначала 2, потом 7.

```
x = int(input())
a=0; b=0
while x > 0:
 if x%2 == 0:
 a += 1
 else:
 b += x%4
 x = x//4
print(a, b)
```

- 112) Укажите наибольшее натуральное число, при вводе которого эта программа напечатает сначала 3, потом 6.

```
x = int(input())
L = 0
M = 0
while x > 0:
 L = L + 1
 if (x % 8) != 0:
 M = M + x % 8
 x = x // 8
print(L)
print(M)
```

- 113) Укажите наибольшее натуральное число, при вводе которого эта программа напечатает сначала 14, потом 3.

```
x = int(input())
L = 0
M = 0
while x > 0:
 M = M + 1
 if x % 2 != 0:
 L = L + x % 8
 x = x // 8
print(L)
print(M)
```

- 114) Укажите наибольшее трёхзначное натуральное число, при вводе которого эта программа напечатает сначала 2, потом 8.

```
x = int(input())
a=0; b=1
while x > 0:
 if x%2 > 0:
 a += x%8
 else:
 b *= x%8
 x = x//8
print(a, b)
```

- 115) Укажите наименьшее натуральное число, при вводе которого эта программа напечатает сначала 2, потом 24.

```
x = int(input())
a=0; b=1
while x > 0:
```

```
if x%2 > 0:
 a += x%8
else:
 b *= x%8
x = x//8
print(a, b)
```

- 116) Укажите наибольшее трёхзначное натуральное число, при вводе которого эта программа напечатает сначала 2, потом 7.

```
x = int(input())
a=0; b=1
while x > 0:
 if x%2 > 0:
 a += x%6
 else:
 b += x%6
 x = x//6
print(a, b)
```

- 117) Укажите наименьшее трёхзначное натуральное число, при вводе которого эта программа напечатает сначала 2, потом 9.

```
x = int(input())
a=0; b=1
while x > 0:
 if x%2 > 0:
 a += x%6
 else:
 b += x%6
 x = x//6
print(a, b)
```

- 118) Укажите наибольшее трёхзначное натуральное число, при вводе которого эта программа напечатает сначала 2, потом 6.

```
x = int(input())
a=0; b=1
while x > 0:
 if x%2 > 0:
 a += 1
 else:
 b += x%5
 x = x//5
print(a, b)
```

- 119) Укажите наименьшее трёхзначное натуральное число, при вводе которого эта программа напечатает сначала 2, потом 9.

```
x = int(input())
a=0; b=1
while x > 0:
 if x%2 > 0:
 a += 1
 else:
 b += x%5
 x = x//5
print(a, b)
```

- 120) (Д.Ф. Муфаззалов, Уфа) Укажите наименьшее четырехзначное натуральное число, при вводе которого приведенная ниже программа напечатает число 4.

```
n = int(input())
a = -1
while n > 9 and a != n % 10:
 a = n % 10
 n //= 10
print(n % 10)
```

- 121) (Д.Ф. Муфаззалов, Уфа) Укажите минимальное натуральное число, при вводе которого приведенная ниже программа напечатает число 4, а потом число 7.

```
n = int(input())
a = -1
k = 0
while n > 9 and a != n % 10:
 a = n % 10
 n //= 10
 k += 1
print(k)
print(a)
```

- 122) (Д.Ф. Муфаззалов, Уфа) Укажите минимальное трехзначное натуральное число, при вводе которого приведенная ниже программа напечатает число 5.

```
n = int(input())
a = -1
while n > 7 and a != n % 8:
 a = n % 8
 n //= 8
if a == n % 8:
 print(a)
else:
 print(n)
```

- 123) (Д.Ф. Муфаззалов, Уфа) Укажите наибольшее трехзначное натуральное число, при вводе которого приведенная ниже программа напечатает число 5.

```
n = int(input())
a = -1
while n > 9 and a != n % 10:
 a = n % 10
 n //= 10
if a == n % 10:
 print(a)
else:
 print(n)
```

- 124) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 4, а потом 7.

```
x = int(input())
a = 0
b = 1
while x > 0:
 if x % 2 > 0:
 a = a + x % 13
 else:
```



```

 b = b * (x % 13)
 x = x // 13
 print(a)
 print(b)

```

- 125) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 5, а потом 2.

```

x = int(input())
a = 0
b = 1
while x > 0:
 if x % 2 == 0:
 a = a + x % 13
 else:
 b = b * (x % 13)
 x = x // 13
print(a)
print(b)

```

- 126) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 2, а потом 9.

```

x = int(input())
a = 0
b = 1
while x > 0:
 if x % 2 > 0:
 a = a + x % 11
 else:
 b = b * (x % 11)
 x = x // 11
print(a)
print(b)

```

- 127) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которого алгоритм печатает сначала 1, а потом 8.

```

x = int(input())
a = 0
b = 1
while x > 0:
 if x % 2 > 0:
 a = a + x % 11
 else:
 b = b * (x % 11)
 x = x // 11
print(a)
print(b)

```

- 128) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наибольшее трёхзначное натуральное число, при вводе которого алгоритм печатает сначала 1, а потом 8.

```

x = int(input())
a = 0

```

```
b = 1
while x > 0:
 if x % 2 > 0:
 a = a + x % 11
 else:
 b = b * (x % 11)
 x = x // 11
print(a)
print(b)
```

- 129) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 1, а потом 9.

```
x = int(input())
a = 0
b = 1
while x > 0:
 if x % 2 == 0:
 a = a + x % 9
 else:
 b = b * (x % 9)
 x = x // 9
print(a)
print(b)
```

- 130) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 3, а потом тоже - 3.

```
x = int(input())
a = 0
b = 1
while x > 0:
 if x % 2 == 0:
 a = a + x % 7
 else:
 b = b * (x % 7)
 x = x // 7
print(a)
print(b)
```

- 131) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 3, а потом 12.

```
x = int(input())
a = 0
b = 1
while x > 0:
 if x % 2 > 0:
 a = a + x % 13
 else:
 b = b * (x % 13)
 x = x // 13
print(a)
print(b)
```

- 132) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее четырёхзначное натуральное число, при вводе которого алгоритм печатает сначала 4, а потом тоже - 4.

```
x = int(input())
a = 0
b = 0
while x > 0:
 if x % 2 > 0:
 a = a + x % 9
 else:
 b = b + x % 9
 x = x // 9
print(a)
print(b)
```

- 133) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наибольшее трёхзначное натуральное число, при вводе которого алгоритм печатает сначала 4, а потом тоже - 4.

```
x = int(input())
a = 0
b = 0
while x > 0:
 if x % 2 == 0:
 a = a + x % 5
 else:
 b = b + x % 5
 x = x // 5
print(a)
print(b)
```

- 134) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 0?

```
x = int(input())
a = 0
b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print(a)
print(b)
```

- 135) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 5?

```
x = int(input())
a = 0
while x > 0:
 a = a + 1
 b = x % 6
 x = x // 6
print(a)
print(b)
```

- 136) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 5, а потом 7?

```
x = int(input())
a = 0
while x > 0:
 a = a + 1
 b = x % 9
 x = x // 9
print(a)
print(b)
```

- 137) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 24?

```
x = int(input())
a = 0
b = 1
while x > 0:
 a = a + 1
 if x % 8 != 1:
 b = b * (x % 8)
 x = x // 8
print(a)
print(b)
```

- 138) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 24?

```
x = int(input())
a = 0
b = 1
while x > 0:
 a = a + 1
 if x % 8 != 0:
 b = b * (x % 8)
 x = x // 8
print(a)
print(b)
```

- 139) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 12?

```
x = int(input())
a = 0
b = 1
while x > 0:
 a = a + 1
 if x % 14 != 0:
 b = b * (x % 14)
 x = x // 14
print(a)
print(b)
```

- 140) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 10?

```
x = int(input())
a = 0
b = 1
while x > 0:
 a = a + 1
 if x % 12 != 0:
 b = b * (x % 12)
 x = x // 12
print(a)
print(b)
```

- 141) (Д. Ф. Муфаззалов, Уфа) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает число, сумма цифр которого равна 15.

```
x = int(input())
L = 0
M = 1
while x > 0:
 L = x % 10 * M + L
 x = x // 10
 M = M * 10
print(L)
```

- 142) (Д. Ф. Муфаззалов, Уфа) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает число, сумма цифр которого равна 15.

```
x = int(input())
L = 0
M = 1
while x > 0:
 L = x % 8 * M + L
 x = x // 8
 M = M * 10
print(L)
```

- 143) (Д. Ф. Муфаззалов, Уфа) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает число 110.

```
x = int(input())
L = 0
M = 1
while x > 0:
 L = x % 16 * M + L
 x = x // 16
 M = M * 10
print(L)
```

- 144) Укажите минимальное двузначное натуральное число, при вводе которого этот алгоритм напечатает число 8.

```
x = int(input())
a = 0; b = 10
while x > 0:
 d = x % 6
 if d > a: a = d
 if d < b: b = d
```

```
x = x // 6
print(a+b)
```

- 145) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает число 11.

```
x = int(input())
a = 0; b = 10
while x > 0:
 d = x % 7
 if d > a: a = d
 if d < b: b = d
 x = x // 7
print(a+b)
```

- 146) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает сначала 7, а потом 12.

```
x = int(input())
a = 0; b = 1
while x > 0:
 if x % 2 > 0:
 a += x % 12
 else:
 b *= x % 12
 x = x // 12
print(a)
print(b)
```

- 147) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает число 18.

```
x = int(input())
a = 0; b = 10
while x > 0:
 d = x % 9
 if d > a: a = d
 if d < b: b = d
 x = x // 9
print(a*b)
```

- 148) (Е. Дзюбс) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , большее, чем 100, при вводе которого алгоритм напечатает 21.

```
x = int(input())
L = 3*x - 6
M = 3*x + 99
while L != M:
 if L > M:
 L = L - M
 else: M = M - L
print(M)
```

- 149) (Е. Дзюбс) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наибольшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 2, а потом 1.

```
x = int(input())
a, b = 0, 0
while x > 0:
 y = x % 10
```

```

 if y > 3:
 a = a + 1
 else:
 b = b - 1
 if y < 8:
 b = b + 1
 x = x // 10
 print(a)
 print(b)

```

150) (Е. Джобс) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ .

Укажите наибольшее натуральное число, при вводе которого алгоритм печатает сначала 13, а потом 3.

```

x = int(input())
a = 0
b = 0
while x > 0:
 x = x // 9
 if x % 2 > 0:
 a = a + x % 9
 b = b + 1
 print(a)
 print(b)

```

151) (Досрочный ЕГЭ-2020) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $L$  и  $M$ . Укажите наибольшее натуральное число, при вводе которого алгоритм печатает сначала 2, а потом 3.

```

x = int(input())
L = 0
M = 0
while x > 0:
 M = M + 1
 if x % 2 != 0:
 L = L + x % 8
 x = x // 8
 print(L)
 print(M)

```

152) (Досрочный ЕГЭ-2020) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $L$  и  $M$ . Укажите наибольшее натуральное число, при вводе которого алгоритм печатает сначала 12, а потом 3.

```

x = int(input())
L = 0
M = 0
while x > 0:
 M = M + 1
 if x % 2 == 0:
 L = L + x % 8
 x = x // 8
 print(L)
 print(M)

```

- 153) (А.М. Кабанов) Ниже приведён алгоритм. Получив на вход число  $x$ , этот алгоритм печатает число  $S$ . Укажите наибольшее число  $x$ , при вводе которого алгоритм печатает 82.

```
x = int(input())
P = 90
S = 6 * (x - x % 22)
K = 0
while P < 181:
 K = K + 1
 P = P + K
 S = S - 2 * K
print(S)
```

- 154) (А.М. Кабанов) Ниже приведён алгоритм. Получив на вход число  $x$ , этот алгоритм печатает число  $K$ . Укажите наименьшее число  $x$ , при вводе которого алгоритм печатает 10.

```
x = int(input())
P = x
S = 10 * x
K = 0
while P < S:
 K = K + 1
 S = S - 2 * K
 P = P + K
print(K)
```

- 155) (А.М. Кабанов) Ниже приведён алгоритм. Получив на вход число  $x$ , этот алгоритм печатает число  $K$ . Укажите наибольшее число  $x$ , при вводе которого алгоритм печатает 5.

```
x = int(input())
K = x - 1
P = 100
S = 340
while P < S:
 K = K + 1
 S = S - 2 * K
 P = P + K
K = K - x
print(K)
```

- 156) (Т.Л. Шамасова) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 11, а потом 3.

```
x = int(input())
a = 0; b = 0
while x > 0:
 x = x // 9
 if x % 2 > 0:
 a = a + x % 9
 else:
 b = b + 1
print(a, b)
```

- 157) Получив на вход натуральное число  $x$ , этот алгоритм печатает одно число. Укажите наименьшее натуральное число, при вводе которого алгоритм печатает 48.

```
x = int(input())
a = 1
while x > 0:
 a *= x % 7
 x = x // 7
```



```
print(a)
```

- 158) Получив на вход натуральное число  $x$ , этот алгоритм печатает одно число. Укажите наименьшее натуральное число, при вводе которого алгоритм печатает 54.

```
x = int(input())
a = 1
while x > 0:
 a *= x % 7
 x = x // 7
print(a)
```

- 159) Получив на вход натуральное число  $x$ , этот алгоритм печатает одно число. Укажите наименьшее натуральное число, при вводе которого алгоритм печатает 40.

```
x = int(input())
a = 1
while x > 0:
 a *= x % 6
 x = x // 6
print(a)
```

- 160) Получив на вход натуральное число  $x$ , этот алгоритм печатает одно число. Укажите наименьшее натуральное число, при вводе которого алгоритм печатает 60.

```
x = int(input())
a = 1
while x > 0:
 a *= x % 9
 x = x // 9
print(a)
```

- 161) (Е. Джобс) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа, L и M. Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 2.

```
x = int(input())
L = 1
M = 0
while x > 0:
 if M < L:
 M = M + 1
 else:
 L = L + M
 x = x // 6
print(L)
print(M)
```

- 162) (Е. Джобс) Ниже записана программа. Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает два числа: L и M. Укажите наибольшее число  $x$ , при вводе которого алгоритм выводит сначала 2, а потом 4.

```
x = int(input())
L = 0
M = 10
while x > 5:
 L = L + 1
 if x % 8 < M:
 M = x % 8
 x = x // 8
print(L)
print(M)
```

- 163) (Е. Джобс) Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает число  $S$ .  
Какое наименьшее число  $x$  необходимо ввести, чтобы в результате работы алгоритма на экран было выведено число, большее 100?

```
x = int(input())
B = x
S = -2
A = 4
while B // 2 > 0:
 if B % 2 == 0:
 S = S + A
 else:
 S = S * 3
 B = B // 2
print(S)
```

- 164) (Е. Джобс) Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает два числа.  
Укажите наименьшее число, при вводе которого программа напечатает сначала 3, затем 4.

```
x = int(input())
A = 0
B = 1
while x > 0:
 if x % 7 > 2:
 A = A + 1
 else:
 B = B * (x % 7)
 x = x // 7
print(A)
print(B)
```

- 165) (Е. Джобс) Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает два числа.  
Укажите наибольшее число, при вводе которого программа напечатает сначала 270, затем 90.

```
x = int(input())
P = 0
S = 10*(x - x % 15)
i = 2
while i < 20:
 S = S - 2*i
 P = P + i
 i = i + 2
print(S)
print(P)
```

- 166) (Е. Джобс) Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает число  $S$ .  
Укажите наименьшее число  $x$ , при вводе которого на экран будет выведено число, большее 100.

```
x = int(input())
S = 1
A = 5
while x // 7 > 0:
 if x % 2 == 0:
 S = S + A
 else:
 S = S * (x % 7)
 x = x // 7
print(S)
```

- 167) (Е. Джобс) Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает число  $S$ . Известно, что в результате работы программы на экран выведено минимально возможное число большее 25. Укажите минимальное число  $x$ , для которого это возможно.

```
x = int(input())
S = 1
A = 11
while x // 7 > 0:
 if x % 7 < 4:
 S = S + A
 else:
 S = S + (x % 7)
 x = x // 7
print(S)
```

- 168) (Е. Джобс) Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает число  $S$ . Укажите наименьшее число  $x$ , при вводе которого на экран будет выведено натуральное число, кратное 100.

```
x = int(input())
S = 5
while x > 0:
 if x % 8 > 4:
 S = S + (x % 8)
 else:
 S = S * (x % 8)
 x = x // 8
print(S)
```

- 169) (Е. Джобс) Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает число  $S$ . Сколько существует чисел  $x$ , не превышающих 500, при вводе которых результате работы программы на экране будет выведено число 13.

```
x = int(input())
S = 0
while x > 0:
 if x % 5 > 0:
 S = S + (x % 5)
 else:
 S = S * (x % 5)
 x = x // 5
print(S)
```

- 170) (Е. Джобс) Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает число  $S$ . Укажите наименьшее число  $x$ , большее 50, при вводе которого на экран будет выведено число 1.

```
x = int(input())
S = 0
while x > 0:
 if x % 2 > 0:
 S = S + (x % 7)
 else:
 S = S - (x % 7)
 x = x // 7
print(S)
```

- 171) (А. Кабанов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа. Укажите наибольшее натуральное число, при вводе которого алгоритм печатает сначала 24, а затем 4.

```
x = int(input())
a = 0
b = 0
```

```

while x > 0:
 a = a + 1
 if x % 2 != 0:
 b = b + 1
 x = x // 2
print(a, b)

```

- 172) (А. Кабанов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа. Укажите наибольшее натуральное число, при вводе которого алгоритм печатает сначала 10, а затем 8.

```

x = int(input())
a = 0
b = 0
while x > 0:
 if x%2 == 0:
 a = a + 1
 else:
 b = b + 1
 x = x // 2
print(a, b)

```

- 173) (А. Кабанов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа. Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 7, а затем 7.

```

x = int(input())
a = 0
b = 0
while x > 0:
 a = a + 1
 if x % 11 > b:
 b = x % 11
 x = x // 11
print(a, b)

```

- 174) (А. Кабанов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа. Укажите наименьшие натуральные значения  $x$  и  $y$ , при вводе которого алгоритм печатает сначала 6, а затем 7. В качестве ответа запишите произведение  $x*y$ .

```

x = int(input())
y = int(input())
a = 0
b = 0
while x > 0 or y>0:
 if x > 0:
 a = a + 1
 if y > 0:
 b = b + 1
 x = x // 2
 y = y // 10
print(a, b)

```

- 175) (А. Кабанов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа. Укажите наименьшие натуральные значения  $x$  и  $y$ , при вводе которого алгоритм печатает сначала 4, а затем 5. В качестве ответа запишите произведение  $x*y$ .

```

x = int(input())
y = int(input())
a = 0
b = 0
while x * y > 0:
 if x > 0:

```

```
a = a + 1
if y > 0 and y%7 > b:
 b = y % 7
x = x // 10
y = y // 7
print(a, b)
```

- 176) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Укажите наименьшее возможное значение  $x$ , при вводе которого программа выведет числа 5 и 12.

```
x = int(input())
m = 0
s = 0
while x > 0:
 d = x % 7
 s += d
 if d > m: m = d
 x = x // 7
print(m, s)
```

- 177) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Укажите наименьшее возможное значение  $x$ , при вводе которого программа выведет числа 3 и 14.

```
x = int(input())
m = 0
s = 0
while x > 0:
 d = x % 7
 s += d
 if d > m: m = d
 x = x // 7
print(m, s)
```

- 178) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Укажите наименьшее возможное значение  $x$ , при вводе которого программа выведет числа 3 и 10.

```
x = int(input())
m = 0
s = 0
while x > 0:
 d = x % 6
 s += d
 if d > m: m = d
 x = x // 6
print(m, s)
```

- 179) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Укажите наименьшее возможное значение  $x$ , при вводе которого программа выведет числа 5 и 16.

```
x = int(input())
m = 0
s = 0
while x > 0:
 d = x % 6
 s += d
 if d > m: m = d
 x = x // 6
```

```
print(m,s)
```

- 180) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Укажите наименьшее возможное значение  $x$ , при вводе которого программа выведет числа 3 и 10.

```
x = int(input())
k = x % 5
a = 0
b = 0
while x > 0:
 d = x % 5
 if d == k:
 a += 1
 b += d
 x //= 5
print(a, b)
```

- 181) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Укажите наименьшее возможное значение  $x$ , при вводе которого программа выведет числа 2 и 15.

```
x = int(input())
k = x % 6
a = 0
b = 0
while x > 0:
 d = x % 6
 if d == k:
 a += 1
 b += d
 x //= 6
print(a, b)
```

- 182) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Укажите наименьшее возможное значение  $x$ , при вводе которого программа выведет числа 4 и 11.

```
x = int(input())
k = x % 7
a = 0
b = 0
while x > 0:
 d = x % 7
 if d == k:
 a += 1
 b += d
 x //= 7
print(a, b)
```

- 183) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Укажите наименьшее возможное значение  $x$ , при вводе которого программа выведет числа 3 и 20.

```
x = int(input())
k = x % 8
a = 0
b = 0
while x > 0:
 d = x % 8
 if d == k:
```

```
a += 1
b += d
x //= 8
print(a, b)
```

- 184) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит результат. Укажите наименьшее значение  $x$ , при вводе которого программа выведет число 64.

```
x = int(input())
a = 3*x + 67
b = 3*x - 61
while a != b:
 if a > b:
 a -= b
 else:
 b -= a
print(a)
```

- 185) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит результат. Укажите наименьшее значение  $x$ , при вводе которого программа выведет число 96.

```
x = int(input())
a = 5*x + 345
b = 5*x - 807
while a != b:
 if a > b:
 a -= b
 else:
 b -= a
print(a)
```

- 186) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит результат. Укажите наименьшее значение  $x$ , при вводе которого программа выведет число 158.

```
x = int(input())
a = 3*x + 71
b = 3*x - 87
while a != b:
 if a > b:
 a -= b
 else:
 b -= a
print(a)
```

- 187) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит результат. Укажите наименьшее значение  $x$ , при вводе которого программа выведет число 34.

```
x = int(input())
a = 3*x - 112
b = 3*x + 58
while a != b:
 if a > b:
 a -= b
 else:
 b -= a
print(a)
```

- 188) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит результат. Укажите наименьшее значение  $x$ , при вводе которого программа выведет число 45.

```
x = int(input())
a = x - 61
b = 3*x - 138
while a != b:
 if a > b:
 a -= b
 else:
 b -= a
print(a)
```

- 189) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит результат. Укажите наименьшее значение  $x$ , при вводе которого программа выведет число 15.

```
x = int(input())
a = 2*x - 91
b = 3*x - 159
while a != b:
 if a > b:
 a -= b
 else:
 b -= a
print(a)
```

- 190) (Е. Джобс) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа  $L$  и  $M$ . Сколько существует натуральных чисел  $x$ , при вводе которых алгоритм печатает 6 и 0?

```
x = int(input())
L, M = 0, 0
while x > 0:
 L = L + 1
 if x % 16 % 2 == 0:
 M = M + 1
 else:
 M = M - 1
 x = x // 16
print(L)
print(M)
```

- 191) (Е. Джобс) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует натуральных чисел, цифры в которых расположены в порядке невозрастания, при вводе которых алгоритм печатает сначала 9, а потом 5.

```
x = int(input())
x = (x - x % 8) * 10
a = 1
b = 0
while x > 0:
 if x % 2 != 0:
 a = a * (x % 4)
 else:
 b = b + (x % 4)
 x = x // 8
print(a)
```



- 192) (Е. Джобс) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа  $L$  и  $M$ . При каком наибольшем значении  $x$  после выполнения программы на экран будет выведено сначала число 3, а затем – 7.

```
x = int(input())
L, M = 0, 0
while x > 12:
 L = L + 1
 x = x // 4
 M = x
 if L > M:
 L, M = M, L
print(L)
print(M)
```

- 193) (А. Богданов) Ниже записан алгоритм. Получив на вход число  $X$ , этот алгоритм печатает одно число. Укажите наименьшее число  $X$ , большее 80, при вводе которого алгоритм печатает число 17.

```
x = int(input())
s = 0
while x > 0:
 s = s + x % 9
 x = x // 3
print(s)
```

- 194) (А. Богданов) Ниже записан алгоритм. Получив на вход число  $K$ , этот алгоритм печатает два числа. Укажите наименьшее число  $K$ , при вводе которого алгоритм печатает числа 13 и 25.

```
k = int(input())
x = 1
y = x
while k < 13:
 k += 1
 if k == 7:
 t = 0
 else:
 t = x + y
 x = y
 y = t
print(k, x)
```

- 195) (А. Кабанов) Ниже записана программа. Получив на вход число  $N$ , эта программа печатает число  $c$ . При каком наибольшем натуральном значении переменной  $N$  программа выведет 9?

```
N = int(input())
c = 0; T = 3; d = 3;
while N != 0:
 N = N - T
 T = T + d
 c = c + 1
 if c % 2 == 0:
 c = c + d
print(c)
```

- 196) (А. Кабанов) Ниже записана программа. Получив на вход число  $N$ , эта программа печатает число  $c$ . При каком наименьшем натуральном значении переменной  $N$  программа выведет 8?

```
N = int(input())
c = 0; T = 3; d = 3;
while N != 0:
 N = N - T
 T = T + d
```

```

 c = c + 1
 if c % 2 != 0:
 c = c + d
 print(c)

```

- 197) (А. Кабанов) Ниже записана программа. Получив на вход число N, эта программа печатает число c. При каком наименьшем натуральном значении переменной N программа выведет 6?

```

N = int(input())
c = 0; T = 5; d = 5;
while N != 200:
 N = N + T
 T = T + d
 c = c + 1
 if c % 2 != 0:
 c = c + d
 print(c)

```

- 198) (А. Кабанов) Ниже записана программа. Получив на вход число T, эта программа печатает число c. При каком наименьшем натуральном значении переменной T программа выведет 8?

```

T = int(input())
c = 0; N = 0; d = 5;
while N != 300:
 N = N + T
 T = T + d
 c = c + 1
 if c % 2 != 0:
 c = c + d
 print(c)

```

- 199) (А. Кабанов) Ниже записана программа. Получив на вход число d, эта программа печатает число c. При каком наибольшем натуральном значении переменной d программа выведет 8?

```

d = int(input())
c = 0; N = 0; T = d;
while N != 144:
 N = N + T
 T = T + d
 c = c + 1
 if c % 2 != 0:
 c = c + 5
 print(c)

```

- 200) Ниже записана программа. Получив на вход число x, эта программа печатает числа a и b. При каком наибольшем натуральном значении переменной x программа выведет сначала 10, а потом 13?

```

x = int(input())
a = 1
b = 0
while x > 0:
 d = x % 9
 a *= d
 b += d
 x //= 9
 print(a, b)

```

- 201) Ниже записана программа. Получив на вход число x, эта программа печатает числа a и b. При каком наибольшем натуральном значении переменной x программа выведет сначала 10, а потом 14?

```

x = int(input())

```

```
a = 1
b = 0
while x > 0:
 d = x % 8
 a *= d
 b += d
 x //= 8
print(a, b)
```

- 202) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 12, а потом 14?

```
x = int(input())
a = 1
b = 3
while x > 0:
 d = x % 7
 a *= d
 if d < 4:
 b += d
 x //= 7
print(a, b)
```

- 203) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 30, а потом 31?

```
x = int(input())
a = 5
b = 20
while x > 0:
 d = x % 6
 a *= d
 if d < 3:
 b += d
 x //= 6
print(a, b)
```

- 204) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 28, а потом 17?

```
x = int(input())
a = 4
b = 10
while x > 0:
 d = x % 11
 a *= d
 if d < 5:
 b += d
 x //= 11
print(a, b)
```

- 205) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 24, а потом 16?

```
x = int(input())
a = 2
b = 3
```

```
while x > 0:
 d = x % 4
 a *= d
 if d < 3:
 b += d
 x //= 4
print(a, b)
```

- 206) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 63, а потом 15?

```
x = int(input())
a = 3
b = 5
w = 9
while x > 0:
 d = x % w
 a *= d
 if d < 5:
 b += d
 x //= w
 w = 17 - w
print(a, b)
```

- 207) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 168, а потом 25?

```
x = int(input())
a = 7
b = 15
w = 6
while x > 0:
 d = x % w
 a *= d
 if d < 3:
 b += d
 x //= w
 w = 11 - w
print(a, b)
```

- 208) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 216, а потом 25?

```
x = int(input())
a = 6
b = 18
w = 9
while x > 0:
 d = x % w
 a *= d
 if d < 3:
 b += d
 x //= w
 w = 16 - w
print(a, b)
```

- 209) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 210, а потом 24?

```
x = int(input())
a = 2
b = 14
w = 9
while x > 0:
 d = x % w
 a *= d
 if d < 5:
 b += d
 x //= w
 w = 15 - w
print(a, b)
```

- 210) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 90, а потом 33?

```
x = int(input())
a = 3
b = 24
w = 8
while x > 0:
 d = x % w
 a *= d
 if d < 5:
 b += d
 x //= w
 w = 11 - w
print(a, b)
```

- 211) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 225, а потом 27?

```
x = int(input())
a = 5
b = 13
w = 7
while x > 0:
 d = x % w
 a *= d
 if d < 5:
 b += d
 x //= w
 w = 12 - w
print(a, b)
```

- 212) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 4, а потом 166?

```
x = int(input())
a = b = 0
while x > 0:
 a += 1
 if x % 2 == 0:
```

```
 b += x % 100
 x //= 10
 print(a, b)
```

- 213) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 3, а потом 126?

```
x = int(input())
a = b = 0
while x > 0:
 a += 1
 if x % 2 == 1:
 b += x % 100
 x //= 10
 print(a, b)
```

- 214) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 8, а потом 146?

```
x = int(input())
a = 5; b = 8
while x > 0:
 a += 1
 if x % 2 == 1:
 b += x % 100
 x //= 10
 print(a, b)
```

- 215) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 7, а потом 255?

```
x = int(input())
a = 3; b = 12
while x > 0:
 a += 1
 if x % 2 == 1:
 b += x % 100
 x //= 10
 print(a, b)
```

- 216) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 7, а потом 18?

```
x = int(input())
a = 4; b = 10
while x > 0:
 a += 1
 if x % 2 == 1:
 b += x % 64
 x //= 8
 print(a, b)
```

- 217) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 10, а потом 79?

```
x = int(input())
a = 14; b = 100
```

```
while x > 0:
 a -= 1
 if x % 2 == 1:
 b -= x % 64
 x //= 8
print(a, b)
```

- 218) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 58, а потом 415?

```
x = int(input())
a = 62; b = 320
while x > 0:
 a -= 1
 if x % 2 == 1:
 b += x % 100
 x //= 10
print(a, b)
```

- 219) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает числа  $a$  и  $b$ . При каком наибольшем натуральном значении переменной  $x$  программа выведет сначала 22, а потом 246?

```
x = int(input())
a = 18; b = 432
while x > 0:
 a += 1
 if x % 2 == 1:
 b -= x % 100
 x //= 10
print(a, b)
```

- 220) (Досрочный ЕГЭ-2022) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наименьшем натуральном значении переменной  $s$  программа выведет сначала 10, а потом 19?

```
s = int(input())
P = 10
Q = 8
K1 = 0
K2 = 0
while s <= 100:
 s = s + P
 K1 = K1 + 1
while s >= Q:
 s = s - Q
 K2 = K2 + 1
K1 += s
K2 += s
print(K1, K2)
```

- 221) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наименьшем натуральном значении переменной  $s$  программа выведет сначала 8, а потом 70?

```
s = int(input())
P = 11
Q = 3
K1 = 0
K2 = 0
```

```

while s <= 200:
 s = s + P
 K1 = K1 + 1
while s >= Q:
 s = s - Q
 K2 = K2 + 1
K1 += s
K2 += s
print(K1, K2)

```

- 222) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наименьшем натуральном значении переменной  $s$  программа выведет сначала 8, а потом 26?

```

s = int(input())
P = 10
Q = 5
K1 = 0
K2 = 0
while s <= 120:
 s = s + P
 K1 = K1 + 1
while s >= Q:
 s = s - Q
 K2 = K2 + 1
K1 += s
K2 += s
print(K1, K2)

```

- 223) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наибольшем натуральном значении переменной  $s$  программа выведет сначала 9, а потом 24?

```

s = int(input())
P = 12
Q = 9
K1 = 0
K2 = 0
while s <= 150:
 s = s + P
 K1 = K1 + 1
while s >= Q:
 s = s - Q
 K2 = K2 + 1
K1 += s
K2 += s
print(K1, K2)

```

- 224) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наибольшем натуральном значении переменной  $s$  программа выведет сначала 4, а потом 38?

```

s = int(input())
P = 19
Q = 4
K1 = 0
K2 = 0
while s <= 134:
 s = s + P

```



```

 K1 = K1 + 1
 while s >= Q:
 s = s - Q
 K2 = K2 + 1
 K1 += s
 K2 += s
 print(K1, K2)

```

225) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наибольшем натуральном значении переменной  $s$  программа выведет сначала 11, а потом 33?

```

s = int(input())
P = 17
Q = 5
K1 = 0
K2 = 0
while s <= 140:
 s = s + P
 K1 = K1 + 1
 while s >= Q:
 s = s - Q
 K2 = K2 + 1
 K1 += s
 K2 += s
 print(K1, K2)

```

226) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. Найдите сумму натуральных значений переменной  $s$ , при которых программа выведет сначала 14, а потом 40?

```

s = int(input())
P = 14
Q = 7
K1 = 0
K2 = 0
while s <= 250:
 s = s + P
 K1 = K1 + 1
 while s >= Q:
 s = s - Q
 K2 = K2 + 1
 K1 += s
 K2 += s
 print(K1, K2)

```

227) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. Найдите сумму натуральных значений переменной  $s$ , при которых программа выведет сначала 12, а потом 75?

```

s = int(input())
P = 15
Q = 3
K1 = 0
K2 = 0
while s <= 220:
 s = s + P
 K1 = K1 + 1
 while s >= Q:

```

```
s = s - Q
K2 = K2 + 1
K1 += s
K2 += s
print(K1, K2)
```

- 228) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. Найдите сумму натуральных значений переменной  $s$ , при которых программа выведет сначала 8, а потом 78?

```
s = int(input())
P = 18
Q = 4
K1 = 0
K2 = 0
while s <= 300:
 s = s + P
 K1 = K1 + 1
while s >= Q:
 s = s - Q
 K2 = K2 + 1
K1 += s
K2 += s
print(K1, K2)
```

- 229) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наименьшем натуральном значении переменной  $s$  программа выведет сначала 36, а потом 335?

```
s = int(input())
P = 13
Q = 4
K1 = 0
K2 = 0
while s != 1280:
 s = s + P
 K1 = K1 + 1
while s != Q + K1:
 s = s - Q
 K2 = K2 + 1
K1 += s
K2 += s
print(K1, K2)
```

- 230) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наименьшем натуральном значении переменной  $s$  программа выведет сначала 19, а потом 204?

```
s = int(input())
P = 23
Q = 7
K1 = 0
K2 = 0
while s != 1350:
 s = s + P
 K1 = K1 + 1
while s != Q + K1:
 s = s - Q
 K2 = K2 + 1
```

```
K1 += s
K2 += s
print(K1, K2)
```

- 231) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. При каком наименьшем натуральном значении переменной  $s$  программа выведет сначала 314, а потом 470?

```
s = int(input())
P = 29
Q = 11
K1 = 0
K2 = 0
while s != 2520:
 s = s + P
 K1 = K1 + 1
while s != Q + K1 + K2:
 s = s - Q
 K2 = K2 + 1
K1 += s
K2 += s
print(K1, K2)
```

- 232) (Е. Дзюбс) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. Сколько существует целых значений  $s$ , при которых после выполнения программы на экран будет выведено два числа 13, а затем 17?

```
P = 12
Q = 8
K1 = K2 = D = 0
s = int(input())
s = s + 21
while s >= 43:
 s -= 43
 D += 1
while s < 0:
 s += 43
 D -= 1
while D <= 100:
 D = D + P
 K1 = K1 + 1
while D >= Q:
 D = D - Q
 K2 = K2 + 1
K1 += D
K2 += D
print(K1, K2)
```

- 233) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 15120. Укажите наибольшее возможное значение числа, выведенного вторым.

```
x = int(input())
a = 1; b = 0
while x > 0:
 d = x % 10
 a *= d
 if d > 5:
 b += d
```

```
x //= 10
print(a, b)
```

- 234) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 5145. Укажите наибольшее возможное значение числа, выведенного вторым.

```
x = int(input())
a = 1; b = 0
while x > 0:
 d = x % 10
 a *= d
 if d > 4:
 b += d
 x //= 10
print(a, b)
```

- 235) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 529 200. Укажите наибольшее возможное значение числа, выведенного вторым.

```
x = int(input())
a = 1; b = 0
while x > 0:
 d = x % 10
 a *= d
 if d > 3:
 b += d
 x //= 10
print(a, b)
```

- 236) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 84 000 000. Укажите наибольшее возможное значение числа, выведенного вторым.

```
x = int(input())
a = 1; b = 0
while x > 0:
 d = x % 10
 a *= d
 if d > 2:
 b += d
 x //= 10
print(a, b)
```

- 237) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 1 360 800 000. Укажите наибольшее возможное значение числа, выведенного вторым.

```
x = int(input())
a = 1; b = 0
while x > 0:
 d = x % 10
 a *= d
 if d > 1:
 b += d
 x //= 10
print(a, b)
```

- 238) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 15 120. Укажите **наименьшее** возможное значение числа, выведенного вторым.

```

x = int(input())
a = 1; b = 0
while x > 0:
 d = x % 10
 a *= d
 if d > 2:
 b += d
 x //= 10
print(a, b)

```

- 239) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 5145. Укажите **наименьшее** возможное значение числа, выведенного вторым.

```

x = int(input())
a = 1; b = 0
while x > 0:
 d = x % 10
 a *= d
 if d > 1:
 b += d
 x //= 10
print(a, b)

```

- 240) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 529 200. Укажите **наименьшее** возможное значение числа, выведенного вторым.

```

x = int(input())
a = 1; b = 0
while x > 0:
 d = x % 10
 a *= d
 if d > 4:
 b += d
 x //= 10
print(a, b)

```

- 241) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 84 000 000. Укажите **наименьшее** возможное значение числа, выведенного вторым.

```

x = int(input())
a = 1; b = 0
while x > 0:
 d = x % 10
 a *= d
 if d > 3:
 b += d
 x //= 10
print(a, b)

```

- 242) Ниже записана программа, которая вводит натуральное число  $x$ , выполняет преобразования, а затем выводит два числа. Известно, что при вводе некоторого  $x$  программа первым вывела число 1 360 800 000. Укажите **наименьшее** возможное значение числа, выведенного вторым.

```

x = int(input())
a = 1; b = 0
while x > 0:
 d = x % 10
 a *= d

```

```

 if d > 4:
 b += d
 x //= 10
 print(a, b)

```

- 243) (Е. Джобс) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает одно число. Найдите минимальное значение  $s$ , при котором после выполнения программы на экран будет выведено 58.

```

s = int(input())
k1 = k2 = 0
while s > 0:
 d = s % 10
 if d > 5:
 s = s - 1
 k1 += 1
 else:
 s = s // 10
 k2 += 1
print(k1 + k2)

```

- 244) (ЕГЭ-2022) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа. Найдите наибольшее значение  $x$ , при котором программы напечатает сначала число 6, а потом снова число 6.

```

x = int(input())
N = 0
S = 0
while x > 0:
 N = N + 1
 S = S + x % 2
 if N % 2 > 0:
 S = S + x % 4
 x = x // 2
print(N)
print(S)

```

- 245) (ЕГЭ-2022) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа. Найдите наибольшее значение  $x$ , при котором программы напечатает сначала число 8, а потом снова число 21.

```

x = int(input())
Q = 6
P = 10
K1 = 0
K2 = 0
while x <= 100:
 K1 = K1 + 1
 x = x + P
while x >= Q:
 K2 = K2 + 1
 x = x - Q
L = x + K1
M = x + K2
print(L)
print(M)

```

- 246) (А. Богданов) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа. Найдите наименьшее число  $x$ , при вводе которого алгоритм печатает числа 9 и 256.

```

x = int(input())

```

```
a = 0
b = 0
while x > 1:
 a = a + 1
 if x % 2 == 0:
 x = x // 2
 else:
 x = 3*x + 1
 if x > b:
 b = x
print(a, b)
```

- 247) (Е. Джобс) Ниже записана программа. Получив на вход число  $s$ , эта программа печатает два числа. Сколько существует значений  $s$ , для которых программы выведет сначала 12, а потом 18?

```
s = int(input())
P = 20
Q = 13
K1 = 0
K2 = 0
while s < 230:
 s = s + P
 K1 = K1 + 1
while s >= Q:
 s = s - Q
 K2 = K2 + 1
print(K1, K2)
```

- 248) (Е. Джобс) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает одно число. Укажите наибольшее число, не превосходящее 1000, после обработки которого на экране будет выведено число 70.

```
x = int(input())
K = 9*x - 57
D = 9*x + 13
while K*D > 0:
 if K > D:
 K = K % D
 else:
 D = D % K
print(K + D)
```