

18 (повышенный уровень, время – 6 мин)

Тема: Динамическое программирование

Что проверяется:

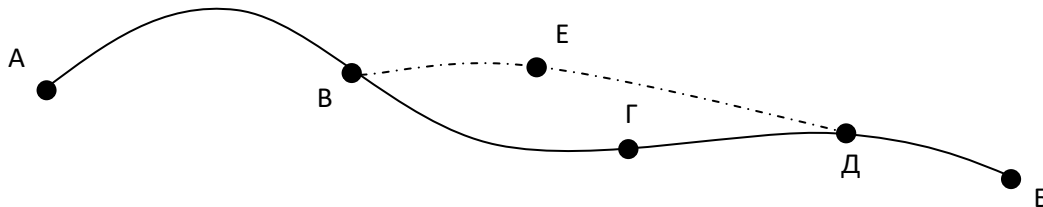
Умение обрабатывать вещественные выражения в электронных таблицах.

3.4.3. Использование инструментов решения статистических и расчётно-графических задач.

1.1.2. Умение представлять и анализировать табличную информацию в виде графиков и диаграмм.

Что нужно знать:

- в задачах, которые предлагаются в этом задании КИМ, нужно найти оптимальный путь для Робота, который перемещается на клетчатом поле. Робот может на каждом шаге выбирать одно из двух направлений движения (например, только вправо и вниз).
- в каждой клетке Робот получает некоторую награду («берёт монету»), и нужно найти такой путь, при котором общая награда будет наибольшая (или наименьшая, если это не награда, а штраф)
- конечно, теоретически можно решить такую задачу полным перебором вариантов: рассмотреть все возможные пути и выбрать лучший. Однако количество возможных путей для полей даже не очень большого размера слишком велико для того, чтобы решить эту задачу за время проведения ЕГЭ, даже если вам удастся безошибочно написать программу для такого перебора.
- эта задача успешно и быстро решается с помощью динамического программирования – метода оптимизации, который предложил американский математик Ричард Беллман. Он сформулировал очень простой принцип оптимальности пути: любая часть оптимального пути оптимальна. Например, пусть мы нашли оптимальный путь из точки А и точку Б, который проходит через точки В, Г и Д:



Принцип Беллмана утверждает, что, например, путь ВГД – это оптимальный путь из В в Д. Если бы это было не так и существовал бы другой, лучший путь между В и Д (например, ВЕД на рисунке), то и путь АВГДБ не был бы оптимальным.

- рассмотрим применение динамического программирования на простом примере: Робот идёт по клетчатому полю из левого верхнего угла в правый нижний; на каждом шаге он может переместиться на одну клетку вправо или на одну клетку вниз. В каждой клетке лежит заданное количество монет:

	А	В	С
1	63	78	58
2	10	1	42
3	25	29	87

Нужно найти такой путь из клетки А1 в клетку С3, пройдя по которому Робот соберёт наибольшее количество монет.

- задачи этого типа решают с конца, то есть сначала предполагают, что Робот стоит в клетке С3, в конечной точке. У него есть единственный вариант – собрать 87 монет, никуда не двигаясь. Построим дополнительную таблицу, в которой для каждой клетки будем записывать наибольшее

число монет, которое может собрать Робот, пройдя из этой клетки в конечную. В правом нижнем углу пишем 87:

	A	B	C
1			
2			
3			87

- рассмотрим клетку B3: стартовав из неё Робот может сделать только один шаг в C3, собрав всего $29 + 87 = 116$ монет; аналогично при старте из C2 робот собирает монеты из ячеек C2 и C3, всего $42 + 87 = 129$:

	A	B	C
1			
2			129
3		116	87

- заметим, что из A3 есть только один возможный путь – через B3, Робот соберёт $25 + 116 = 141$ монету; аналогично из C1 есть только путь через C2, Робот соберёт $58 + 129 = 187$ монет:

	A	B	C
1			187
2			129
3	141	116	87

- если Робот стартует из клетки B2, у него есть выбор: пойти вправо (набрав $1+129=130$ монет) или вниз ($1+116=117$ монет); очевидно, что первый вариант лучше, так что в эту ячейку записываем 130 (как лучшую из двух сумм)

	A	B	C
1			187
2		130	129
3	141	116	87

- аналогично в каждую из остальных ячеек записываем лучшую из двух сумм, которые можно получить при движении вправо и вниз:

	A	B	C
1	328	265	187
2	151	130	129
3	141	116	87

- наибольшая сумма, которую может собрать Робот, двигаясь из левого верхнего угла в правый нижний, находится в ячейке A1 рабочей таблицы, она равна 328
- по рабочей таблице можно восстановить путь Робота. Например, выясним, куда пойдёт Робот на первом шаге. Мы знаем, что 328 получено как сумма количества монет в A1 (63) и одной из соседних ячеек B1 или A2. В ячейке, куда должен перейти Робот, должно быть число $328 - 63 = 265$, это ячейка B1.

Пример задания:

P-02 (А. Кабанов). Дана последовательность натуральных чисел. Рассматриваются всевозможные пары чисел, находящиеся на расстоянии не менее 3 чисел друг от друга. В каком количестве таких пар сумма чисел чётна. Исходная последовательность записана в виде одной строки электронной таблицы.

Решение (электронные таблицы):

- 1) Рассмотрим решение на примере следующей таблицы.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	45	92	74	27	76	86	70	85	75	55	39	64	39	29

- 2) Для определённости будем двигаться справа налево. Для каждого числа a_i рассмотрим набор из предыдущих чисел (кроме последних трёх). Число a_i будет образовывать подходящие пары с числами такой же чётности. Для каждого числа вычислим остаток от деления на 2. В ячейке **A2** запишем формулу **=ОСТАТ (A1 ; 2)** и скопируем её на все остальные ячейки второй строки.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	45	92	74	27	76	86	70	85	75	55	39	64	39	29
2	1	0	0	1	0	0	0	1	1	1	1	0	1	1

- 3) С помощью формулы **СЧЁТЕСЛИ** подсчитаем, сколько из предыдущих чисел (кроме последних трёх) имеют такую же чётность. Например, в ячейке **K3** формула будет иметь вид **=СЧЁТЕСЛИ (\$A\$2 : G2 ; "=" & K2)**. Начало диапазона – абсолютная ссылка для корректного автозаполнения. Формулу запишем в ячейку **K3** и скопируем во все ячейки третьей строки. Для первых 4 чисел количество пар равно 0.
- 4) Общее количество равно сумме пар, образуемых каждым числом. В ячейке **A4** запишем формулу **=СУММ (A3 : N3)**

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	45	92	74	27	76	86	70	85	75	55	39	64	39	29
2	1	0	0	1	0	0	0	1	1	1	1	0	1	1
3	0	0	0	0	0	1	2	2	2	2	2	5	4	5
4	25													

- 5) Ответ: **25**.

Пример задания:

P-01 (А. Кабанов). Дана последовательность натуральных чисел. Рассматриваются всевозможные пары чисел, находящиеся на расстоянии не более 4 чисел друг от друга. В каком количестве таких пар сумма чисел меньше 100. Исходная последовательность записана в виде одной строки электронной таблицы.

Решение (электронные таблицы):

- 1) Рассмотрим решение на примере следующей таблицы.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	45	92	74	27	76	86	70	85	75	55	39	64	39	29

- 2) Для определённости будем двигаться справа налево. Для каждого числа a_i рассмотрим набор из пяти предыдущих чисел (если таковые существуют). Число a_i будет образовывать подходящие пары с числами, меньшими $100 - a_i$. С помощью формулы **СЧЁТЕСЛИ** подсчитаем, сколько из предыдущих пяти чисел меньше $100 - a_i$. Например, в ячейке **F2** формула будет иметь вид **=СЧЁТЕСЛИ (A1 : E1 ; "<" & (100 - F1))**. Формулу запишем в ячейку **F2** и скопируем во все ячейки строки. Для первых 5 чисел формулу необходимо откорректировать.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	45	92	74	27	76	86	70	85	75	55	39	64	39	29
2	0	0	0	1	0	0	1	0	0	0	1	0	2	4

- 3) Общее количество равно сумме пар, образуемых каждым числом. В ячейке **A3** запишем формулу **=СУММ (A2 : N2)**

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	45	92	74	27	76	86	70	85	75	55	39	64	39	29
2	0	0	0	1	0	0	1	0	0	0	1	0	2	4
3	9													

- 4) Ответ: **9**.

Ещё пример задания:

Р-00 (демо-2021). Квадрат разлинован на $N \times N$ клеток ($1 < N < 17$). Исполнитель Робот может перемещаться по клеткам, выполняя за одно перемещение одну из двух команд: вправо или вниз. По команде вправо Робот перемещается в соседнюю правую клетку, по команде вниз – в соседнюю нижнюю. При попытке выхода за границу квадрата Робот разрушается. Перед каждым запуском Робота в каждой клетке квадрата лежит монета достоинством от 1 до 100. Посетив клетку, Робот забирает монету с собой; это также относится к начальной и конечной клетке маршрута Робота. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой верхней клетки в правую нижнюю. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.

Исходные данные записаны в файле **18-0.xls** в виде электронной таблице размером $N \times N$, каждая ячейка которой соответствует клетке квадрата.

Решение (электронные таблицы):

- 1) откроем электронную таблицу и увидим следующую таблицу размером 10 на 10:

	A	B	C	D	E	F	G	H	I	J	K
1	51	21	93	48	45	100	67	39	18	29	
2	57	43	97	51	92	10	93	32	19	58	
3	63	16	31	16	78	88	90	72	37	67	
4	10	57	64	25	96	50	81	65	91	69	
5	99	43	95	7	40	76	18	34	5	65	
6	35	19	71	77	64	38	62	56	10	2	
7	100	57	27	26	51	33	100	11	53	1	
8	11	79	49	46	37	69	80	31	25	39	
9	22	71	20	23	11	12	39	16	64	34	
10	4	25	87	84	30	48	77	13	40	33	
11											

Робот начинает движение из верхнего левого угла (из ячейки A1) и перемещается в правый нижний (то есть в J10)

- 2) при использовании метода динамического программирования требуется выделить для вычислений дополнительную таблицу такого же размера; проще всего сделать так:
- скопировать исходную таблицу вниз
 - обвести её рамкой (и/или выделить фоном), чтобы запомнить исходный размер;
 - стереть все данные в копии.

вот что должно получиться:

	A	B	C	D	E	F	G	H	I	J
1	51	21	93	48	45	100	67	39	18	29
2	57	43	97	51	92	10	93	32	19	58
3	63	16	31	16	78	88	90	72	37	67
4	10	57	64	25	96	50	81	65	91	69
5	99	43	95	7	40	76	18	34	5	65
6	35	19	71	77	64	38	62	56	10	2
7	100	57	27	26	51	33	100	11	53	1
8	11	79	49	46	37	69	80	31	25	39
9	22	71	20	23	11	12	39	16	64	34
10	4	25	87	84	30	48	77	13	40	33
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										

- 3) дальше мы будем работать только с областью, выделенной жёлтым фоном

- 4) предположим, что робот уже находится в правом нижнем углу; в этом случае он может получить только сумму в этой ячейке, то есть величину J10; записываем в J22 формулу =J10; после ввода формулы видим в ячейке J22 значение 33, как и ожидалось:

	A	B	C	D	E	F	G	H	I	J
4	10	57	64	25	96	50	81	65	91	69
5	99	43	95	7	40	76	18	34	5	65
6	35	19	71	77	64	38	62	56	10	2
7	100	57	27	26	51	33	100	11	53	1
8	11	79	49	46	37	69	80	31	25	39
9	22	71	20	23	11	12	39	16	64	34
10	4	25	87	84	30	48	77	13	40	33
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										33

- 5) рассмотрим нижний ряд: если Робот находится в одной из ячеек последней строки 10, то он может идти только вправо, собирая монеты в последней строке; например, начав движение из ячейки I10 он соберёт монету в этой ячейке и во всех (в данном случае – в одной) следующих, то есть формула в ячейке I22 должна быть =I10+J22; обратите внимание, что первая ссылка в формуле (I10) обращается к исходной таблице (берёт из неё одно значение), а вторая (J22) – к рабочей (берёт всё накопленное далее); неудивительно, что в I22 появляется число 73:

	A	B	C	D	E	F	G	H	I	J
1	51	21	93	48	45	100	67	39	18	29
2	57	43	97	51	92	10	93	32	19	58
3	63	16	31	16	78	88	90	72	37	67
4	10	57	64	25	96	50	81	65	91	69
5	99	43	95	7	40	76	18	34	5	65
6	35	19	71	77	64	38	62	56	10	2
7	100	57	27	26	51	33	100	11	53	1
8	11	79	49	46	37	69	80	31	25	39
9	22	71	20	23	11	12	39	16	64	34
10	4	25	87	84	30	48	77	13	40	33
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22									73	33

- 6) эту формулу копируем (протаскиваем за маркер заполнения) по всей строке 22:

	A	B	C	D	E	F	G	H	I	J
1	51	21	93	48	45	100	67	39	18	29
2	57	43	97	51	92	10	93	32	19	58
3	63	16	31	16	78	88	90	72	37	67
4	10	57	64	25	96	50	81	65	91	69
5	99	43	95	7	40	76	18	34	5	65
6	35	19	71	77	64	38	62	56	10	2
7	100	57	27	26	51	33	100	11	53	1
8	11	79	49	46	37	69	80	31	25	39
9	22	71	20	23	11	12	39	16	64	34
10	4	25	87	84	30	48	77	13	40	33
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22	441	437	412	325	241	211	163	86	73	33

здесь хорошо видно, что для того чтобы сумма накапливалась, вторая ссылка в формуле должна указывать на строку 22, а не на строку 10

- 7) аналогично если Робот находится в последнем столбце, он может двигаться только вниз, собирая по пути все монеты; вводим в ячейку **J21** формулу **=J9+J22** и протаскиваем (копируем) её вверх на весь столбец **J** вспомогательной таблицы:

	A	B	C	D	E	F	G	H	I	J
1	51	21	93	48	45	100	67	39	18	29
2	57	43	97	51	92	10	93	32	19	58
3	63	16	31	16	78	88	90	72	37	67
4	10	57	64	25	96	50	81	65	91	69
5	99	43	95	7	40	76	18	34	5	65
6	35	19	71	77	64	38	62	56	10	2
7	100	57	27	26	51	33	100	11	53	1
8	11	79	49	46	37	69	80	31	25	39
9	22	71	20	23	11	12	39	16	64	34
10	4	25	87	84	30	48	77	13	40	33
11										
12										
13										397
14										368
15										310
16										243
17										174
18										109
19										107
20										106
21										67
22	441	437	412	325	241	211	163	86	73	33

- 8) займёмся центральными ячейками жёлтой таблицы, которые пока не заполнены; пусть Робот находится в ячейке **I9**; тогда для того, чтобы получить максимальную сумму, ему нужно выбрать лучший из двух путей – пойти в **I10** или в **J9**; из второй таблицы видим, что в первом случае дополнительно к значению **I9** он получит сумму 73, а во втором – только 67. поэтому выгоднее первый вариант; в формуле нужно выбрать максимум из значений ячеек **I10** и **J9**, получаем **=I9+МАКС (I22 ; J21)** ; обратите внимание, что оба аргумента функции **МАКС** находятся в рабочей таблице

	A	B	C	D	E	F	G	H	I	J
1	51	21	93	48	45	100	67	39	18	29
2	57	43	97	51	92	10	93	32	19	58
3	63	16	31	16	78	88	90	72	37	67
4	10	57	64	25	96	50	81	65	91	69
5	99	43	95	7	40	76	18	34	5	65
6	35	19	71	77	64	38	62	56	10	2
7	100	57	27	26	51	33	100	11	53	1
8	11	79	49	46	37	69	80	31	25	39
9	22	71	20	23	11	12	39	16	64	34
10	4	25	87	84	30	48	77	13	40	33
11										
12										
13										397
14										368
15										310
16										243
17										174
18										109
19										107
20										106
21										67
22	441	437	412	325	241	211	163	86	73	33

- 9) для всех оставшихся ячеек принцип вычисления максимальной суммы тот же самый: нужно добавить к значению этой ячейки в исходной таблице максимум из накопленных сумм, которые Робот собирает в случае двух возможных шагов; копируем формулу из I21 на весь диапазон A13 : I21 (сначала можно растянуть формулу на диапазон-столбец I13 : I21, а затем этот диапазон – на нужный диапазон A13 : I21)

Первый ответ к задаче – это максимальная сумма, накопленная при движении из левого верхнего угла; она записана в левом верхнем углу рабочей таблицы, то есть в ячейке A13 (она выделена зелёным фоном):

13	1204	1153	1132	990	938	893	793	542	415	397
14	1139	1082	1039	942	891	736	726	503	390	368
15	1004	926	884	815	799	721	633	471	371	310
16	941	910	853	729	704	608	543	399	334	243
17	931	832	789	630	598	558	462	316	230	174
18	779	713	694	623	546	482	444	282	225	109
19	744	644	519	492	466	415	382	226	215	107
20	598	587	483	434	388	351	282	193	162	106
21	530	508	432	348	252	223	202	153	137	67
22	441	437	412	325	241	211	163	86	73	33

- 10) чтобы найти наименьшую возможную сумму, нужно в формуле в п. 9 заменить функцию МАКС на МИН: =I9+МИН (I22 ; J21)

13	502	451	532	439	537	495	395	328	289	397
14	487	430	468	391	492	400	396	303	271	368
15	450	387	371	340	466	390	347	295	252	310
16	455	445	388	324	388	302	257	223	215	243
17	493	402	394	299	292	252	176	158	124	174
18	394	359	354	336	316	275	237	175	119	109
19	411	340	283	259	252	255	244	144	160	107
20	311	335	256	233	201	222	213	133	131	106
21	300	278	207	187	164	153	141	102	131	67
22	441	437	412	325	241	211	163	86	73	33

- 11) Ответ: 1204 502

- 12) заметим, что решение никак не зависит от того, что таблица квадратная; этот метод так же хорошо работает и для любых прямоугольных таблиц

- 13) видеоразбор решения этой задачи сделал А. Сидоров

(<https://www.youtube.com/watch?v=xoKzzj5QX18>)

- 14) можно обойтись вообще одной формулой, если добавить к рабочей таблице дополнительно нулевую строку внизу и нулевой столбец справа (Д.Ф. Муфаззалов, М.В. Кузнецова):

12											
13	1204	1153	1132	990	938	893	793	542	415	397	0
14	1139	1082	1039	942	891	736	726	503	390	368	0
15	1004	926	884	815	799	721	633	471	371	310	0
16	941	910	853	729	704	608	543	399	334	243	0
17	931	832	789	630	598	558	462	316	230	174	0
18	779	713	694	623	546	482	444	282	225	109	0
19	744	644	519	492	466	415	382	226	215	107	0
20	598	587	483	434	388	351	282	193	162	106	0
21	530	508	432	348	252	223	202	153	137	67	0
22	441	437	412	325	241	211	163	86	73	33	0
23	0	0	0	0	0	0	0	0	0	0	

15) в этом случае во все ячейки можно скопировать ту формулу, которую мы внесли в ячейку I21

16) поскольку функции **МАКС** и **МИН** игнорируют пустые ячейки, записывать в дополнительные ячейки нули вообще **не нужно!**;

17) удобнее всего вписать в ячейку J22, соответствующую конечной точке маршрута, аналогичную формулу **=J10+МАКС (J23 ; K22)**

и затем скопировать её во все остальные ячейки рабочей таблицы

18) (**М.В. Кузнецова**) если вам удобнее заполнять таблицу слева направо и сверху вниз, можно найти стоимость обратного маршрута (из правой нижней ячейки в левую верхнюю), ведь в данной задаче они одинаковы; для этого освобождаем строку выше и столбец левее рабочей таблицы и вводим в её левый верхний угол формулу **=A1+МАКС (A13 ; B12)** ; эту формулу копируем на всю таблицу, и считываем ответ в правом нижнем углу рабочей таблицы:

	A	B	C	D	E	F	G	H	I	J	K
1	51	21	93	48	45	100	67	39	18	29	
2	57	43	97	51	92	10	93	32	19	58	
3	63	16	31	16	78	88	90	72	37	67	
4	10	57	64	25	96	50	81	65	91	69	
5	99	43	95	7	40	76	18	34	5	65	
6	35	19	71	77	64	38	62	56	10	2	
7	100	57	27	26	51	33	100	11	53	1	
8	11	79	49	46	37	69	80	31	25	39	
9	22	71	20	23	11	12	39	16	64	34	
10	4	25	87	84	30	48	77	13	40	33	
11											
12											
13	51	72	165	213	258	358	425	464	482	511	
14	57	151	262	313	405	415	518	550	569	627	
15	63	174	293	329	483	571	661	733	770	837	
16	10	174	329	379	483	571	661	733	770	837	
17	99	174	329	379	483	571	661	733	770	837	
18	35	315	342	523	600	683	743	822	897	913	1034
19	100	415	472	550	626	734	776	922	933	986	1035
20	11	426	551	600	672	771	845	1002	1033	1058	1097
21	22	448	622	642	695	782	857	1041	1057	1122	1156
22	4	452	647	734	818	848	905	1118	1131	1171	1204

Решение (программа):

1) в наборе файлов к этому заданию должен быть файл с расширением **.csv** (англ. *comma separated values* – значения, разделённые запятыми); чаще всего в качестве разделителя используется точка с запятой; если такого файла нет, его можно всегда получить, сохранив электронную таблицу в формате CSV; в данной задаче файл 18-0.csv выглядит так:

```

51;21;93;48;45;100;67;39;18;29
57;43;97;51;92;10;93;32;19;58
63;16;31;16;78;88;90;72;37;67
10;57;64;25;96;50;81;65;91;69
99;43;95;7;40;76;18;34;5;65
35;19;71;77;64;38;62;56;10;2
100;57;27;26;51;33;100;11;53;1
11;79;49;46;37;69;80;31;25;39
22;71;20;23;11;12;39;16;64;34
4;25;87;84;30;48;77;13;40;33

```


- 2) напишем программу на языке Python, поскольку на других языках она получится значительно сложнее и использование такого подхода становится, мягко говоря, не очень обоснованно
- 3) сначала читаем данные из файла в матрицу (двухмерный массив) **data**, попутно определяем размеры матрицы: число строк **N** и число столбцов **M**:

```
data = []
for s in open( "18-0.csv" ):
    row = list( map( int, s.split(';') ) )
    data.append( row )
N = len( data )
M = len( data[0] )
```

- 4) строим рабочую матрицу такого же размера, что и исходная; заполняем её нулями:
- 5) заполняем ячейку в правом нижнем углу (конечная точка маршрута Робота); учитываем, что нумерация строк и столбцов матрицы в Python начинается с нуля, так что эта ячейка имеет индексы **N-1, M-1**:

```
work[N-1][M-1] = data[N-1][M-1]
```

- 6) заполняем последнюю строку (с индексом **N-1**) справа налево:

```
for col in range(M-2, -1, -1):
    work[N-1][col] = data[N-1][col] + work[N-1][col+1]
```

- 7) заполняем последний столбец (с индексом **M-1**) снизу вверх:

```
for row in range(N-2, -1, -1):
    work[row][M-1] = data[row][M-1] + work[row+1][M-1]
```

- 8) заполняем центральную часть (сверху вниз, справа налево):

```
for row in range(N-2, -1, -1):
    for col in range(M-2, -1, -1):
        work[row][col] = data[row][col] + \
            max(work[row][col+1], work[row+1][col])
```

в данном случае программа будет искать максимальную сумму; для поиска минимальной суммы нужно заменить функцию **max** на **min**

- 9) остаётся вывести ответ – значение левого верхнего угла рабочей матрицы **work**, то есть значение ячейки с нулевыми индексами:

```
print( work[0][0] )
```

- 10) приведём полную программу:

```
data = []
for s in open( "18-0.csv" ):
    row = list( map( int, s.split(';') ) )
    data.append( row )
N = len( data )
M = len( data[0] )

work = [ [0]*M for i in range(N) ]
work[N-1][M-1] = data[N-1][M-1]

for col in range(M-2, -1, -1):
    work[N-1][col] = data[N-1][col] + work[N-1][col+1]

for row in range(N-2, -1, -1):
    work[row][M-1] = data[row][M-1] + work[row+1][M-1]

for row in range(N-2, -1, -1):
    for col in range(M-2, -1, -1):
```

```

        work[row][col] = data[row][col] + max(work[row][col+1],
work[row+1][col])

```

```

print( work[0][0] )

```

11) ещё раз обратим внимание на то, что такой способ решения задачи рекомендуется только тем, кто значительно лучше владеет программированием, чем электронными таблицами

12) (Муфаззалов Д.Ф., г. Уфа) Программу можно сократить, если использовать одну формулу, содержащую элементы справа и снизу данного элемента, для всех элементов матрицы. Рассмотрим сначала случай поиска минимальной суммы. Так как у элементов на последней строке и в последнем столбце нет элементов снизу и справа соответственно, добавим к матрице один столбец справа и одну строку снизу. Элемент на последней строке в последнем столбце матрицы work должен быть равен такому же элементу матрицы data, поэтому элементы справа и снизу от него должны быть равны нулю. Остальные элементы матрицы work на последней строке и последнем столбце не должны зависеть от элементов снизу и справа соответственно, поэтому там должны быть числа, не влияющие на выбор значения, то есть гораздо большие чем другое число. Поместим туда сумму элементов матрицы.

```

data = []
ma = 0
for s in open( "18-0.csv" ):
    row = list( map( int, s.split(' ',') ) )
    data.append( row )
    ma += sum(row)
N = len( data )+1
M = len( data[0] )+1
work = [ [ma]*M for i in range(N) ]
work[N-2][M-1], work[M-1][N-2] = 0, 0
for row in range(N-2,-1,-1):
    for col in range(M-2,-1,-1):
        work[row][col] = data[row][col] + min(work[row][col+1],
work[row+1][col])
print( work[0][0] )

```

Для случая поиска максимальной суммы все дополнительные элементы сделаем нулевыми:

```

data = []
ma = 0
for s in open( "18-0.csv" ):
    row = list( map( int, s.split(' ',') ) )
    data.append( row )
    ma += sum(row)
N = len( data )+1
M = len( data[0] )+1
work = [ [0]*M for i in range(N) ]
work[N-2][M-1], work[M-1][N-2] = 0, 0
for row in range(N-2,-1,-1):
    for col in range(M-2,-1,-1):
        work[row][col] = data[row][col] + max(work[row][col+1],
work[row+1][col])
print( work[0][0] )

```

Решение (программа на Java, М. Коротков):

1) полный текст программы:

```

import java.io.IOException;

```

```

import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.Arrays;
public class Main {
    public static void main(String[] args) throws IOException {
        // считываем данные из файла в двумерный массив
        Integer[][] data = Files.lines(Paths.get("18-0.csv"))
            .map(line -> Arrays.stream(line.split(";"))
                .map(Integer::parseInt)
                .toArray(Integer[]::new))
            .toArray(Integer[][]::new);
        final int N = data.length;
        final int M = data[0].length;
        int[][] calc = new int[N][M];
        calc[0][0] = data[0][0];
        // заполняем первую строку
        for (int j = 1; j < M; j++)
            calc[0][j] = data[0][j] + calc[0][j-1];
        // заполняем первый столбец
        for (int i = 1; i < N; i++)
            calc[i][0] = data[i][0] + calc[i-1][0];
        // заполняем оставшиеся ячейки
        for (int i = 1; i < N; i++)
            for (int j = 1; j < M; j++)
                // для нахождения мин. суммы Math.max
                // заменить на Math.min
                calc[i][j] = data[i][j] +
                    Math.max(calc[i-1][j], calc[i][j-1]);
        System.out.println("Ответ: " + calc[N-1][M-1]);
    }
}

```

Задачи для тренировки:

- [illegible]

Робот, пройдя из левой верхней клетки в правую нижнюю. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.

- 13) Исходные данные записаны в файле **18-6.xls** в виде электронной таблицы прямоугольной формы. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой НИЖНЕЙ клетки в правую ВЕРХНЮЮ. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 14) Исходные данные записаны в файле **18-7.xls** в виде электронной таблицы прямоугольной формы. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой верхней клетки в правую нижнюю. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 15) Исходные данные записаны в файле **18-7.xls** в виде электронной таблицы прямоугольной формы. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой НИЖНЕЙ клетки в правую ВЕРХНЮЮ. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 16) Исходные данные записаны в файле **18-8.xls** в виде электронной таблицы прямоугольной формы. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой верхней клетки в правую нижнюю. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 17) Исходные данные записаны в файле **18-8.xls** в виде электронной таблицы прямоугольной формы. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой НИЖНЕЙ клетки в правую ВЕРХНЮЮ. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 18) Исходные данные записаны в файле **18-9.xls** в виде электронной таблицы прямоугольной формы. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой верхней клетки в правую нижнюю. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 19) Исходные данные записаны в файле **18-9.xls** в виде электронной таблицы прямоугольной формы. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой НИЖНЕЙ клетки в правую ВЕРХНЮЮ. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 20) Исходные данные записаны в файле **18-10.xls** в виде электронной таблицы прямоугольной формы. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой верхней клетки в правую нижнюю. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 21) Исходные данные записаны в файле **18-10.xls** в виде электронной таблицы прямоугольной формы. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой НИЖНЕЙ клетки в правую ВЕРХНЮЮ. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 22) **(В.Н. Шубинкин, г. Казань) В любой клетке может быть стена (стены обозначены значениями больше 100, но меньше 500).** При попытке зайти на клетку со стеной Робот разрушается. Исходные данные записаны в файле **18-11.xls** в виде электронной таблицы прямоугольной формы. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой верхней клетки в правую нижнюю, не разрушившись. Известно, что такой путь существует. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 23) **(В.Н. Шубинкин, г. Казань) В любой клетке может быть яма (ямы обозначены значениями меньше 0, но больше -400).** При попытке зайти на такую клетку Робот застревает в яме и не может двигаться дальше.
Исходные данные записаны в файле **18-12.xls** в виде электронной таблицы прямоугольной формы. Определите максимальную и минимальную денежную сумму, которую может собрать

Робот, пройдя из левой верхней клетки в правую нижнюю, не застряв в яме. Известно, что такой путь существует. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.

- 24) (В.Н. Шубинкин, г. Казань) В любой клетке поля может быть стена (стены обозначены значениями больше 100, но меньше 500) или яма (ямы обозначены значениями меньше 0, но больше -400). При попытке зайти на клетку со стеной Робот разрушается. При попытке зайти на клетку с ямой Робот застревает в ней и не может двигаться дальше.

Исходные данные записаны в файле 18-13.xls в виде электронной таблицы прямоугольной формы. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой верхней клетки в правую нижнюю, не разрушившись и не застряв в яме. Известно, что такой путь существует. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.

- 25) (Е. Джобс) Квадрат разлинован на $N \times N$ клеток ($1 < N < 17$). Исполнитель Робот может перемещаться по клеткам, выполняя за одно перемещение одну из двух команд: вправо или вниз. По команде вправо Робот перемещается в соседнюю правую клетку, по команде вниз – в соседнюю нижнюю. При попытке выхода за границу квадрата Робот разрушается. Перед каждым запуском Робота в каждой клетке квадрата записано число от 10 до 99. Посетив клетку с нечетным значением, Робот увеличивает счет на 1; иначе увеличивает счёт на 2.

Определите максимальное и минимальное значение счета, который может собрать Робот, пройдя из левой верхней клетки в правую нижнюю. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.

Исходные данные представляют собой электронную таблицу в файле 18-J1 размером $N \times N$, каждая ячейка которой соответствует клетке квадрата.

Пример входных данных:

1	8	8	4
10	1	1	3
1	3	12	2
2	3	5	6

Для указанных входных данных ответом должна быть пара чисел 9 12.

- 26) (Е. Джобс) Квадрат разлинован на $N \times N$ клеток ($3 < N < 15$), где N – нечетное число. На поле работает 4 исполнителя Грузовичок, которые начинают движение из центральной клетки. Например, для $N = 5$ из клетки С3. Каждый исполнитель двигается в один из углов – левый верхний, правый верхний, левый нижний или правый нижний – и может двигаться соответственно только – налево и вверх, направо и вверх, вниз и влево, вниз и вправо.

Исполнители работают независимо друг от друга на своей копии поля. Каждая пройденная клетка содержит число – массу в килограммах забираемого груза. Цель исполнителя – забрать как можно больший объем груза, выраженный в килограммах. Необходимо найти наилучшие результаты работы каждого Грузовичка. В ответе запишите четыре числа в порядке возрастания.

Исходные данные представляют собой электронную таблицу в файле 18-J2 размером $N \times N$, каждая ячейка которой соответствует клетке квадрата.

Пример входных данных:

1	8	8	4	10
10	1	1	3	2
1	3	12	2	8
2	3	5	6	11
5	19	14	11	5

Для указанных входных данных ответом должна быть четверка чисел – результаты работы четырех исполнителей в порядке возрастания:

30 35 55 47

- 27) (Е. Джобс) Квадрат разлинован на $N \times N$ клеток ($3 < N < 15$). В каждой клетке записано целое число. На поле работает исполнитель Контур, который суммирует все клетки вокруг клетки, в которой находится. Для клеток, находящихся на краю квадрата, находится сумма значений клеток, которые лежат внутри квадрата. Например, для ячейки A1 нужно найти сумму B1, A2, B2. Необходимо найти минимальный и максимальный результаты работы исполнителя Контур в заданном поле. Исходные данные представляют собой электронную таблицу в файле 18–J3 размером $N \times N$, каждая ячейка которой соответствует клетке квадрата.

Пример входных данных:

1	8	8	4	10
10	1	1	3	2
1	3	12	2	8
2	3	5	6	11
5	19	14	11	5

Для указанных входных данных ответом должна быть пара чисел – минимальное и максимальное значения: **9 70**

- 28) (Е. Джобс) Квадрат разлинован на $N \times N$ клеток ($3 < N < 17$). В каждой клетке записано целое число. Исполнитель Робот может перемещаться по клеткам, выполняя за одно перемещение одну из двух команд: вправо или вниз. По команде вправо Робот перемещается в соседнюю правую клетку, по команде вниз – в соседнюю нижнюю. При попытке выхода за границу квадрата Робот разрушается. В каждой клетке квадрата записано число от 10 до 99 или 0. Посетив клетку, Робот прибавляет к счету значение, записанное в этой клетке. Необходимо найти максимальный и минимальный результаты работы исполнителя Контур в заданном поле. Запрещается посещать одну клетку дважды, а также клетки с нулевым значением. Известно, что как минимум один путь из начальной клетки в конечную точно существует. Исходные данные представляют собой электронную таблицу в файле 18–J4 размером $N \times N$, каждая ячейка которой соответствует клетке квадрата.

Пример входных данных:

1	8	8	4	10
10	1	1	0	2
1	3	12	0	8
2	0	0	0	11
5	19	14	11	5

Для указанных входных данных ответом должна быть пара чисел – минимальное и максимальное значения: **57 68**

- 29) (Е. Джобс) Квадрат разлинован на $N \times N$ клеток ($3 < N < 17$). В каждой клетке лежат конфеты, количество которых соответствует записанному числу. На поле работает исполнитель Дружище, который съедает все конфеты в клетке. Также, если исполнитель проходит между двумя четными или двумя нечетными значениями, то Добрый Волшебник дает ему еще 10 конфет, которые он, конечно же, сразу съедает. Так, например, если исполнитель приходит в клетку C3 из клетки B3, считается, что он прошел между клетками C2 и C4, если в C3 из C2 – между B3 и D3. Исполнитель может перемещаться по клеткам, выполняя за одно перемещение одну из двух команд: вправо или вниз. По команде вправо Дружище перемещается в соседнюю правую клетку, по команде вниз – в соседнюю нижнюю. При попытке выхода за границу квадрата Дружище расстраивается, что ему не дают конфеты, и отказывается идти дальше. Нам важно, чтобы Дружище съел как можно меньше конфет и при этом добрался из левой верхней клетки в правую нижнюю. Исходные данные представляют собой электронную таблицу в файле 18–J5 размером $N \times N$, каждая ячейка которой соответствует клетке квадрата.

Пример входных данных:

1	8	8	4	10
10	1	1	3	2
1	3	12	2	8
2	3	5	6	11
5	19	14	11	5

Для указанных входных данных ответом должно быть число – минимально возможное съедаемое количество конфет – 53.

- 30) Дана последовательность вещественных чисел. Из неё необходимо выбрать несколько подряд идущих чисел так, чтобы каждое следующее число было меньше предыдущего. Какую максимальную сумму могут иметь выбранные числа?

Например, для входных данных

3, 3
5, 2
5, 9
1, 3
1, 7
4, 5

максимально возможная сумма равна 7,2, в ответе надо записать число 7.

Исходные данные записаны в виде столбца электронной таблицы в файле **18-14.xls**.

- 31) Дана последовательность вещественных чисел. Из неё необходимо выбрать несколько подряд идущих чисел так, чтобы каждое следующее число было больше предыдущего. Какую максимальную сумму могут иметь выбранные числа?

Например, для входных данных

3, 3
5, 2
5, 9
1, 3
1, 7
4, 5

максимально возможная сумма равна 14,4, в ответе надо записать число 14.

Исходные данные записаны в виде столбца электронной таблицы в файле **18-14.xls**.

- 32) Дана последовательность вещественных чисел. Из неё необходимо выбрать несколько подряд идущих чисел так, чтобы каждое следующее число было меньше предыдущего. Какую максимальную сумму могут иметь выбранные числа?

Исходные данные записаны в виде столбца электронной таблицы в файле **18-15.xls**.

- 33) Дана последовательность вещественных чисел. Из неё необходимо выбрать несколько подряд идущих чисел так, чтобы каждое следующее число было больше предыдущего. Какую максимальную сумму могут иметь выбранные числа?

Исходные данные записаны в виде столбца электронной таблицы в файле **18-15.xls**.

- 34) Дана последовательность вещественных чисел. Из неё необходимо выбрать несколько подряд идущих чисел так, чтобы каждое следующее число было меньше предыдущего. Какую максимальную сумму могут иметь выбранные числа?

Исходные данные записаны в виде столбца электронной таблицы в файле **18-16.xls**.

- 35) Дана последовательность вещественных чисел. Из неё необходимо выбрать несколько подряд идущих чисел так, чтобы каждое следующее число было больше предыдущего. Какую максимальную сумму могут иметь выбранные числа?

Исходные данные записаны в виде столбца электронной таблицы в файле **18-16.xls**.

- 36) Дана последовательность вещественных чисел. Из неё необходимо выбрать несколько подряд идущих чисел так, чтобы каждое следующее число было меньше предыдущего. Какую максимальную сумму могут иметь выбранные числа?
Исходные данные записаны в виде столбца электронной таблицы в файле **18-17.xls**.
- 37) Дана последовательность вещественных чисел. Из неё необходимо выбрать несколько подряд идущих чисел так, чтобы каждое следующее число было больше предыдущего. Какую максимальную сумму могут иметь выбранные числа?
Исходные данные записаны в виде столбца электронной таблицы в файле **18-17.xls**.
- 38) Дана последовательность вещественных чисел. Из неё необходимо выбрать несколько подряд идущих чисел так, чтобы каждое следующее число было меньше предыдущего. Какую максимальную сумму могут иметь выбранные числа?
Исходные данные записаны в виде столбца электронной таблицы в файле **18-18.xls**.
- 39) Дана последовательность вещественных чисел. Из неё необходимо выбрать несколько подряд идущих чисел так, чтобы каждое следующее число было больше предыдущего. Какую максимальную сумму могут иметь выбранные числа?
Исходные данные записаны в виде столбца электронной таблицы в файле **18-18.xls**.
- 40) **(А. Кабанов)** Дана последовательность натуральных чисел. Из неё необходимо выбрать последовательность подряд идущих чисел так, чтобы каждое число было нечётным. Какую максимальную длину может иметь выбранная последовательность?
Исходные данные записаны в виде столбца электронной таблицы в файле **18-k1.xls**.
- 41) **(А. Кабанов)** Дана последовательность натуральных чисел. Из неё необходимо выбрать несколько подряд идущих чисел так, чтобы каждое число было чётным. Какую максимальную сумму могут иметь выбранные числа?
Исходные данные записаны в виде столбца электронной таблицы в файле **18-k1.xls**.
- 42) **(А. Кабанов)** Дана таблица вещественных чисел размера $N \times N$ ($1 < N \leq 20$). Перемещаться между числами можно по горизонтали и вертикали (в любом направлении). Необходимо найти самую длинную последовательность чисел, такую, что каждое следующее число больше предыдущего. В ответе запишите длину этой цепочки.
Исходные данные записаны в виде электронной таблицы в файле **18-k2.xls**.
- 43) **(А. Кабанов)** Дана таблица вещественных чисел размера $N \times N$ ($1 < N \leq 20$). Перемещаться между числами можно по горизонтали и вертикали (в любом направлении). Необходимо выбрать несколько подряд идущих чисел, таких, что каждое следующее число больше предыдущего. Какую максимальную сумму могут иметь выбранные числа?
Исходные данные записаны в виде электронной таблицы в файле **18-k2.xls**.
- 44) **(А. Кабанов)** Дана последовательность натуральных чисел. Рассматриваются всевозможные пары чисел, находящиеся на расстоянии не более 4 чисел друг от друга. Определите количество таких пар, для которых сумма чисел меньше 100. Исходные данные записаны в виде столбца электронной таблицы в файле **18-k3.xls**.
- 45) **(А. Кабанов)** Дана последовательность натуральных чисел. Рассматриваются всевозможные пары чисел, находящиеся на расстоянии не более 5 чисел друг от друга. Определите количество таких пар, для которых сумма чисел чётная. Исходные данные записаны в виде столбца электронной таблицы в файле **18-k3.xls**.
- 46) **(А. Кабанов)** Дана последовательность натуральных чисел. Рассматриваются всевозможные пары чисел, находящиеся на расстоянии не более 4 чисел друг от друга. Определите количество таких пар, для которых сумма чисел находится в диапазоне от 1000 до 1500, не включая 1000 и 1500. Исходные данные записаны в виде столбца электронной таблицы в файле **18-k3.xls**.
- 47) **(А. Кабанов)** Дана последовательность натуральных чисел. Рассматриваются всевозможные пары чисел, находящиеся на расстоянии не менее 10 чисел друг от друга. Определите количество таких

пар, для которых сумма чисел меньше 200. Исходные данные записаны в виде столбца электронной таблицы в файле **18-k3.xls**.

- 48) (А. Кабанов) Дана последовательность натуральных чисел. Рассматриваются всевозможные пары чисел, находящиеся на расстоянии не менее 8 чисел друг от друга. Определите количество таких пар, для которых сумма чисел нечётная. Исходные данные записаны в виде столбца электронной таблицы в файле **18-k3.xls**.
- 49) (А. Кабанов) Дана последовательность натуральных чисел. Рассматриваются всевозможные пары чисел, находящиеся на расстоянии не менее 6 чисел друг от друга. Определите количество таких пар, для которых сумма чисел находится в диапазоне от 1500 до 2000, включая 1500 и 2000. Исходные данные записаны в виде столбца электронной таблицы в файле **18-k3.xls**.
- 50) (А. Кабанов) Дана последовательность натуральных чисел. Рассматриваются всевозможные пары чисел, находящиеся на расстоянии не более 2 чисел друг от друга. Определите максимальную сумму чисел среди таких пар. Исходные данные записаны в виде столбца электронной таблицы в файле **18-k3.xls**.
- 51) (А. Кабанов) Дана последовательность натуральных чисел. Рассматриваются всевозможные пары чисел, находящиеся на расстоянии не более 4 чисел друг от друга. Определите минимальную чётную сумму среди таких пар. Исходные данные записаны в виде столбца электронной таблицы в файле **18-k3.xls**.
- 52) (А. Кабанов) Дана последовательность натуральных чисел. Наибольшей возрастающей подпоследовательностью называется сама длинная подпоследовательность элементов, простирающаяся снизу вверх и такая, что каждый следующий элемент больше предыдущего. Найдите длину наибольшей возрастающей подпоследовательности. Исходные данные записаны в виде столбца электронной таблицы в файле **18-k3.xls**.
- 53) (В.Н. Шубинкин) Исходные данные для Робота (см. задачу Р-00) записаны в файле **18-0.xls** в виде электронной таблицы прямоугольной формы. Робот может брать монеты **только с тех клеток, где количество монет чётно**. Если количество монет нечётно, то Робот не берёт в этой клетке ни одной монеты. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой верхней клетки в правую нижнюю. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 54) (В.Н. Шубинкин) Исходные данные для Робота (см. задачу Р-00) записаны в файле **18-0.xls** в виде электронной таблицы прямоугольной формы. Робот может брать монеты **только с тех клеток, где количество монет чётно**. Если количество монет нечётно, то Робот не берёт в этой клетке ни одной монеты. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой НИЖНЕЙ клетки в правую ВЕРХНЮЮ. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 55) (В.Н. Шубинкин) Исходные данные для Робота (см. задачу Р-00) записаны в файле **18-0.xls** в виде электронной таблицы прямоугольной формы. Робот может брать монеты **только с тех клеток, где количество монет нечётно**. Если количество монет чётно, то Робот не берёт в этой клетке ни одной монеты. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой верхней клетки в правую нижнюю. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 56) (В.Н. Шубинкин) Исходные данные для Робота (см. задачу Р-00) записаны в файле **18-0.xls** в виде электронной таблицы прямоугольной формы. Робот может брать монеты **только с тех клеток, где количество монет нечётно**. Если количество монет чётно, то Робот не берёт в этой клетке ни одной монеты. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой НИЖНЕЙ клетки в правую ВЕРХНЮЮ. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 57) (В.Н. Шубинкин) Исходные данные для Робота (см. задачу Р-00) записаны в файле **18-1.xls** в виде электронной таблицы прямоугольной формы. Робот может брать монеты **только с тех кле-**

[illegible]

неров, **полностью заполненных** монетами. Если контейнер не заполнен до конца, а монеты в клетке кончились, робот высыпает из него монеты перед переходом в следующую клетку. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой верхней клетки в правую нижнюю. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.

- 66) (В.Н. Шубинкин) Исходные данные для Робота записаны в файле **18-0.xls** в виде электронной таблицы прямоугольной формы. С каждой клетки Робот забирает наибольшее количество контейнеров вместимостью 8 монет каждый, **полностью заполненных** монетами. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой НИЖНЕЙ клетки в правую ВЕРХНЮЮ. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 67) (В.Н. Шубинкин) Исходные данные для Робота записаны в файле **18-1.xls** в виде электронной таблицы прямоугольной формы. С каждой клетки Робот забирает наибольшее количество контейнеров вместимостью 8 монет каждый, **полностью заполненных** монетами. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой верхней клетки в правую нижнюю. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 68) (В.Н. Шубинкин) Исходные данные для Робота записаны в файле **18-1.xls** в виде электронной таблицы прямоугольной формы. С каждой клетки Робот забирает наибольшее количество контейнеров вместимостью 8 монет каждый, **полностью заполненных** монетами. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой НИЖНЕЙ клетки в правую ВЕРХНЮЮ. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 69) (В.Н. Шубинкин) Исходные данные для Робота записаны в файле **18-2.xls** в виде электронной таблицы прямоугольной формы. С каждой клетки Робот забирает наибольшее количество контейнеров вместимостью 8 монет каждый, **полностью заполненных** монетами. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой верхней клетки в правую нижнюю. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 70) (В.Н. Шубинкин) Исходные данные для Робота записаны в файле **18-2.xls** в виде электронной таблицы прямоугольной формы. С каждой клетки Робот забирает наибольшее количество контейнеров вместимостью 8 монет каждый, **полностью заполненных** монетами. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой НИЖНЕЙ клетки в правую ВЕРХНЮЮ. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 71) (В.Н. Шубинкин) Исходные данные для Робота записаны в файле **18-11.xls** в виде электронной таблицы прямоугольной формы. В любой клетке **может быть стена (стены обозначены значениями больше 100, но меньше 500)**. При попытке зайти на клетку со стеной Робот разрушается. С каждой клетки Робот забирает все монеты, если их количество кратно 3 или 4 (иначе он не берёт ни одной монеты). Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой верхней клетки в правую нижнюю. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.
- 72) (В.Н. Шубинкин) Исходные данные для Робота записаны в файле **18-11.xls** в виде электронной таблицы прямоугольной формы. В любой клетке **может быть стена (стены обозначены значениями больше 100, но меньше 500)**. При попытке зайти на клетку со стеной Робот разрушается. С каждой клетки Робот забирает все монеты, если их количество кратно 3 или 4 (иначе он не берёт ни одной монеты). Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой НИЖНЕЙ клетки в правую ВЕРХНЮЮ. В ответе укажите два числа – сначала максимальную сумму, затем минимальную.