

Информатика. 8 класс

Решения

1 вариант

№	Правильный ответ	Балл																																																																																					
1.	<p>Вычислим количество миллисекунд в сутках: $24 * 60 * 60 * 1000 = 86400000$</p> <p>Вычислим количество миллисекунд, за которое сохраняется один фрагмент файла размером в 100 КБ: $35 + 5 + 30 = 70$</p> <p>Вычислим максимальное количество фрагментов, которое можно передать за сутки: $86400000 / 70 \sim 1234285,7143$</p> <p>Так как число фрагментов – число целое, то округлим максимальное количество фрагментов до 1234285.</p> <p>Вычислим объём этих фрагментов в ГБайт (с округлением в меньшую сторону): $1234285 * 100 \text{ КБайт} = 123428500 \text{ КБайт} \sim 120\,535 \text{ МБайт} \sim 117 \text{ ГБайт}$</p> <p>Ответ: 1234285 фрагментов или 123428500 Кбайт = 117 ГБ</p>	15																																																																																					
2.	<p>Запишем таблицу истинности для правой части функции (Бетта(Бетта(Бетта(X_2, X_3), X_4), X_5))):</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>X_2</th> <th>X_3</th> <th>X_4</th> <th>X_5</th> <th>(Бетта(Бетта(Бетта(X_2, X_3), X_4), X_5))</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p>Для X_2, X_3, X_4, X_5, существуют 5 комбинаций значений, при которых часть функции (Бетта(Бетта(Бетта(X_2, X_3), X_4), X_5)) имеет значение «0». Если выражение (X_0 и X_1) имеет значение «1» (это возможно в единственном случае, когда $X_0=1$ и $X_1=1$), то ложность выражения в целом может быть обеспечена только ложностью второй части</p>	X_2	X_3	X_4	X_5	(Бетта(Бетта(Бетта(X_2, X_3), X_4), X_5))	0	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0	1	1	1	0	1	0	0	1	0	1	0	1	1	0	1	1	0	0	0	1	1	1	1	1	0	0	0	0	1	0	0	1	1	1	0	1	0	0	1	0	1	1	1	1	1	0	0	1	1	1	0	1	1	1	1	1	0	0	1	1	1	1	1	20
X_2	X_3	X_4	X_5	(Бетта(Бетта(Бетта(X_2, X_3), X_4), X_5))																																																																																			
0	0	0	0	1																																																																																			
0	0	0	1	1																																																																																			
0	0	1	0	0																																																																																			
0	0	1	1	1																																																																																			
0	1	0	0	1																																																																																			
0	1	0	1	1																																																																																			
0	1	1	0	0																																																																																			
0	1	1	1	1																																																																																			
1	0	0	0	0																																																																																			
1	0	0	1	1																																																																																			
1	0	1	0	0																																																																																			
1	0	1	1	1																																																																																			
1	1	0	0	1																																																																																			
1	1	0	1	1																																																																																			
1	1	1	0	0																																																																																			
1	1	1	1	1																																																																																			

	<p>выражения, и, следовательно, есть 5 комбинаций значений переменных, которые будут давать значение «0». Если выражение (X0 и X1) имеет значение «0» (отметим, что это возможно в трёх случаях), то вторая часть выражения может принимать любое значение. Во второй части выражения 4 переменных, следовательно, есть 2^4 комбинаций их значений. Тогда еще существует $3 \cdot 2^4 = 48$ комбинаций значений переменных, которые будут давать значение «0». Следовательно, всего $5 + 48 = 53$ комбинаций</p> <p>Ответ: 53 комбинаций</p>	
3.	<p>Остатки от деления на 5: 0, 1, 2, 3, 4. Остатки от деления на 3: 0, 1, 2. Всего возможно 15 комбинаций: (0 0), (1 1), (2 2), (0 3), (1 4), (2 0), (0 1), (1 2), (2 3), (0 4), (1 0), (2 1), (0 2), (1 3), (2 4) Итого за 15 итераций цикла получим 45. 1 000 000 000 000 = 66 666 666 666 раз по 15 итераций + 10 итераций. За 10 итераций (0 0), (1 1), (2 2), (0 3), (1 4), (2 0), (0 1), (1 2), (2 3), (0 4) в сумме получим 29 $45 * 66\ 666\ 666\ 666 + 29 = 2\ 999\ 999\ 999\ 999$</p> <p>Ответ: 2 999 999 999 999</p>	15
4.	<pre> n = int(input()) a = [0]*n for i in range(n): a[i] = int(input()) fl = True for i in range(2,n): if (a[i-2] + a[i-1]) != a[i]: fl = False break if fl: print((a[-1] + a[-2])+(a[-1] + a[-2]+a[-1]) +(a[-1] + a[-2]+a[-1]+a[-1] + a[-2])) else: print('NO') </pre>	20
5.	<pre> from math import sqrt N = 10 x = [] def dist(x, y): return sqrt((x[0]-y[0])**2+(x[1]-y[1])**2) for i in range(N): a = [int(i) for i in input().split(' ')] x.append((a[0],a[1])) Smax = -1 for i in range(N): for j in range(i+1,N): for m in range(j+1,N): for n in range(m+1,N): if dist(x[i],x[j]) == dist(x[i],x[m]) and dist(x[i],x[j]) == </pre>	30

```

dist(x[n],x[m]) and dist(x[i],x[n]) == dist(x[j],x[m]) and (dist(x[i],x[j]) *
dist(x[i],x[m])>Smax):
    Smax = dist(x[i],x[j]) * dist(x[i],x[m])
print(Smax)

```

2 вариант

1. Вычислим количество миллисекунд в сутках:
 $24 * 60 * 60 * 1000 = 86\,400\,000$
 Вычислим количество миллисекунд, за которое сохраняется один фрагмент файла размером в 250 КБ:
 $60 + 5 + 30 = 95$
 Вычислим максимальное количество фрагментов, которое можно передать за сутки:
 $86400000 / 95 \sim 909\,473,6842$
 Так как число фрагментов – число целое, то округлим максимальное количество фрагментов до 909 473.
 Вычислим объём этих фрагментов в ГБайт (с округлением в меньшую сторону):
 $909\,473 * 250 \text{ КБайт} = 227\,368\,250 \text{ КБайт} \sim 222\,039 \text{ МБайт} \sim 216 \text{ Гбайт}$
 Ответ: 909 473 фрагментов или 227 368 250 КБайт

2. Запишем таблицу истинности для правой части функции $((X_2 \downarrow X_3) \downarrow (X_4 \downarrow X_5))$:

X_2	X_3	X_4	X_5	$((X_2 \downarrow X_3) \downarrow (X_4 \downarrow X_5))$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Часть функции $((X_2 \downarrow X_3) \downarrow (X_4 \downarrow X_5))$ имеет значение «0» в 7 комбинациях. Если выражение $(X_0$ или $X_1)$ имеет значение «1» (3 случая), то ложность выражения может быть только в случае ложности второй части выражения, и, следовательно, есть 7 комбинаций значений переменных. Если выражение $(X_0$ или $X_1)$ имеет значение «0»

	<p>(это возможно в одном случае), то вторая часть выражения может принимать любое значение. Во второй части выражения 4 переменных, следовательно, есть 2^4 комбинаций их значений. Тогда еще существует $2^4 = 16$ комбинаций значений переменных, которые будут давать значение «0». Следовательно, всего $3*7 + 16 = 37$ комбинаций</p> <p>Ответ: 37 комбинаций</p>	
3.	<p>Остатки от деления на 13: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12</p> <p>Остатки от деления на 3: 0, 1, 2.</p> <p>Всего возможно 39 комбинаций:</p> <p>(0 0) (2 2) (1 4) (0 6) (2 8) (1 10) (0 12) (2 1) (1 3) (0 5) (2 7) (1 9) (0 11) (2 0) (1 2) (0 4) (2 6) (1 8) (0 10) (2 12) (1 1) (0 3) (2 5) (1 7) (0 9) (2 11) (1 0) (0 2) (2 4) (1 6) (0 8) (2 10) (1 12) (0 1) (2 3) (1 5) (0 7) (2 9) (1 11)</p> <p>Так как только при четных i происходит вычитания, то за 78 итераций цикла от S отнимем 273.</p> <p>$999\ 999\ 999\ 999 = 12\ 820\ 512\ 820$ раз по 78 итераций + 39 итераций</p> <p>То есть последние 39 итераций будут такими: (0 0) (2 2) (1 4) (0 6) (2 8) (1 10) (0 12) (2 1) (1 3) (0 5) (2 7) (1 9) (0 11) (2 0) (1 2) (0 4) (2 6) (1 8) (0 10) (2 12), в сумме эти остатки равны 140</p> <p>$1\ 000\ 000\ 000\ 000\ 000 - (12\ 820\ 512\ 820 * 273 + 140) =$ $996\ 500\ 000\ 000\ 000$</p> <p>Ответ: 996 500 000 000 000</p>	15
4.	<pre>n = int(input()) a = [0]*n a[0], a[1], a[2] = 2, 2, 2 if 7 < n < 20: for i in range(3, n): a[i] = a[i-1] + a[i-3] print(a[n-1])</pre>	20
5.	<pre>from math import sqrt N = 10 x = [] def dist(x, y): return sqrt((x[0]-y[0])**2+(x[1]-y[1])**2) for i in range(N): a = [int(i) for i in input().split(' ')] x.append((a[0], a[1])) Smax = -1 for i in range(N): for j in range(i+1, N): for m in range(j+1, N): for n in range(m+1, N): if dist(x[i], x[j]) == dist(x[i], x[m]) and dist(x[i], x[j]) == dist(x[n], x[m]) and dist(x[i], x[n]) == dist(x[j], x[m]) and (dist(x[i], x[j]) * dist(x[i], x[m])) > Smax: Smax = dist(x[i], x[j]) * dist(x[i], x[m])</pre>	30

print(Smax)

3 вариант

1. Вычислим количество фрагментов в файле: с округлением в большую сторону, так как весь файл необходимо передать
 $120 \cdot 1024 \cdot 1024 \text{ КБ} / 100 \text{ КБ} \approx 1\,258\,292$
Вычислим количество миллисекунд, за которое загружается один фрагмент файла размером в 100 КБ:
 $45 + 10 + 45 = 100$
Вычислим количество миллисекунд, необходимое для загрузки файла:
 $1\,258\,292 \cdot 100 = 125\,829\,200$ миллисекунд
Найдем количество часов: (с округлением в большую сторону):
 $125\,829\,200 / 1000 \approx 125\,830$ секунд $\approx 2\,098$ минут ≈ 35 часов
Ответ: 35 часов

2. Запишем таблицу истинности для правой части функции $((X_2 | X_3) | X_4) | X_5$:

X_2	X_3	X_4	X_5	$((X_2 X_3) X_4) X_5$
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Для X_2, X_3, X_4, X_5 , существуют 5 комбинаций значений, при которых часть функции $((X_2 | X_3) | X_4) | X_5$ имеет значение «0». Если выражение $(X_0$ или $X_1)$ имеет значение «1» (это возможно в трех случаях), то ложность выражения в целом может быть обеспечена только ложностью второй части выражения, и, следовательно, есть 5 комбинаций значений переменных, которые будут давать значение «0». Если выражение $(X_0$ или $X_1)$ имеет значение «0» (отметим, что это возможно в одном случае), то вторая часть выражения может принимать любое значение. Во второй части выражения 4 переменных, следовательно, есть 2^4 комбинаций их значений. Тогда еще существует $2^4 = 16$ комбинаций значений переменных, которые будут давать

	значение «0». Следовательно, всего $3*5 + 16 = 31$ комбинация Ответ: 31 комбинация	
3.	<p>Остатки от деления на 10: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9</p> <p>Остатки от деления на 6: 0, 1, 2, 3, 4, 5</p> <p>Остатки от деления на 5: 0, 1, 2, 3, 4</p> <p>Так как только при i катных 4 происходят вычисления, то за каждые 60 итераций возможно встретить 15 комбинаций: для вычисления значения d:</p> <p>(0 0 0) $d=0$, (4 4 4) $d=-12$, (8 2 3) $d=2$, (2 0 2) $d=2$, (6 4 1) $d=2$, (0 2 0) $d=0$, (4 0 4) $d=4$, (8 4 3) $d=-4$, (2 2 2) $d=-2$, (6 0 1) $d=6$, (0 4 0) $d=0$, (4 2 4) $d=-4$, (8 0 3) $d=8$, (2 4 2) $d=-6$, (6 2 1) $d=4$</p> <p>За каждые 60 итераций цикла к S прибавится по 0. 999 999 999 999= 16 666 666 666 раз по 60 итераций + 39 итераций. То есть за последние 39 итераций остатки и значение d будут: (0 0 0) $d=0$, (4 4 4) $d=-12$, (8 2 3) $d=2$, (2 0 2) $d=2$, (6 4 1) $d=2$, (0 2 0) $d=0$, (4 0 4) $d=4$, (8 4 3) $d=-4$, (2 2 2) $d=-2$, (6 0 1) $d=6$, в сумме d равны -2</p> <p>Ответ: -2</p>	15
4.	<pre>n = int(input()) a = [0]*n a[0], a[1] = 3,3 for i in range(2,n): a[i] = a[i-1] + a[i-2] print(a[n-1])</pre>	20
5.	<pre>from math import sqrt N = 10 x = [] def dist(x, y): return sqrt((x[0]-y[0])**2+(x[1]-y[1])**2) for i in range(N): a = [int(i) for i in input().split(' ')] x.append((a[0],a[1])) Smax = -1 for i in range(N): for j in range(i+1,N): for m in range(j+1,N): for n in range(m+1,N): if dist(x[i],x[j]) == dist(x[i],x[m]) and dist(x[i],x[j]) == dist(x[n],x[m]) and dist(x[i],x[n]) == dist(x[j],x[m]) and (dist(x[i],x[j]) * dist(x[i],x[m])>Smax): Smax = dist(x[i],x[j]) * dist(x[i],x[m]) print(Smax)</pre>	30
4 вариант		
1.	Будем считать объем аудиофайла по формуле $I = f*i*t*k$, где I – общий объем аудиофайла, f – частота дискретизации, i – глубина кодирования(запишем в байтах), t – время в секундах, а k – число	15

каналов.

$$i = 24 \text{ бит} = 3 \text{ байта}$$

$$I = 32000 * 3 * 120 * 2 = 2^{12} * 5^4 * 3^2 \text{ байт} = 23\,040\,000 \text{ байт}$$

$$\text{На флеш-карте } 2 \text{ МБайта} = 2^{11} \text{Кбайта} = 2^{21} \text{ байт}$$

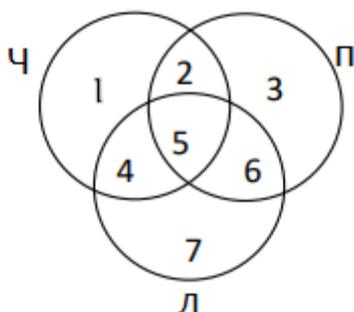
Тогда ответом на нашу задачу является результат разности размера аудиофайла и объёма флеш-карты: $2^{12} * 5^4 * 3^2 - 2^{21} = 2^{12} * (5\,625 - 2^9)$

$$= 2^{12} * (5\,625 - 512) = 4\,096 * 5\,113 = 20\,942\,848 \text{ байт}$$

Ответ: 20 942 848 байт

2. Сделаем замену, Чеснок -> А, Перец -> В, Лук -> С. Таким образом получаем:

Запрос	Найдено страниц (в тысячах)
Ч	153
Ч & П & Л	47
Ч П	221
П	164
Ч & Л	75
Ч П Л	251
Л	137



$$\text{Ч} = 1+2+4+5 = 153$$

$$\text{Ч \& П \& Л} = 5 = 47$$

$$\text{Ч | П} = 1+3 + 4+ 6 + 2+5 = 221$$

$$\text{П} = 2+3+5+6 = 164$$

$$\text{Ч \& Л} = 4+5 = 75$$

$$\text{Ч | П | Л} = 1+2+3+4+5+6+7 = 251$$

$$\text{Л} = 4+5+6+7 = 137$$

$$1 = 29$$

$$2 = 49$$

$$3 = 36$$

$$4 = 28$$

$$5 = 47$$

$$6 = 32$$

$$7 = 30$$

$$\text{Ч \& П} = 2+5 = \text{П} + \text{Ч} - \text{Ч | П} = 96$$

$$\text{П | Л} = 2+3 + 4+5+6+7 = 222$$

Раскроем по формуле включения\исключения запросы Ч | П | Л, Ч | П и искомый П | Л получаем:

$$\text{Ч | П | С} = \text{Ч} + \text{П} + \text{Л} - \text{Ч \& П} - \text{Ч \& Л} - \text{П \& Л} + \text{Ч \& П \& Л}$$

	<p> $\text{Ч} \text{П} = \text{Ч} + \text{П} - \text{Ч}\&\text{П} \Rightarrow \text{Ч}\&\text{П} = \text{Ч} + \text{П} - \text{Ч} \text{П}$ $\text{П} \text{Л} = \text{П} + \text{Л} - \text{П}\&\text{Л} \Rightarrow \text{П}\&\text{Л} = \text{П} + \text{Л} - \text{П} \text{Л}$ Подставим выраженные значения в запрос $\text{Ч} \text{П} \text{Л}$: $\text{Ч} \text{П} \text{Л} = \text{Ч} + \text{П} + \text{Л} - \text{Ч} - \text{П} + \text{Ч} \text{П} - \text{Ч}\&\text{Л} - \text{П} - \text{Л} + \text{П} \text{Л} + \text{Ч}\&\text{П}\&\text{Л} =$ $\text{Ч} \text{П} - \text{Ч}\&\text{Л} - \text{П} + \text{П} \text{Л} + \text{Ч}\&\text{П}\&\text{Л}$ Теперь обозначим $\text{П} \text{Л}$ за X, как искомое. Подставим в уравнение $\text{Ч} \text{П} \text{Л}$ все количества запросов с нужными знаками, получаем: $251 = 221 - 75 - 164 + X + 47$ $251 = 29 + X$ $X = 222$ Ответ: 222 тыс. страниц </p>	
3.	<p> Остатки от деления на 10: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 Остатки от деления на 3: 0, 1, 2. Учитывая, что d должно быть четным, всего возможно 15 комбинаций: (0 0) $d=0$, (1 1) $d=0$, (2 2) $d=-2$, (6 0) $d=6$, (7 1) $d=6$, (8 2) $d=4$, (2 0) $d=2$, (3 1) $d=2$, (4 2) $d=0$, (8 0) $d=8$, (9 1) $d=8$, (0 2) $d=-4$, (4 0) $d=4$, (5 1) $d=4$, (6 2) $d=2$ За каждые 30 итераций цикла S будет увеличиваться на 40 $1\ 000\ 000\ 000\ 000 = 33\ 333\ 333\ 333$ раз по 30 итераций + 10 итераций. За последние 10 итераций (0 0) $d=0$, (1 1) $d=0$, (2 2) $d=-2$, (6 0) $d=6$, (7 1) $d=6$, (8 2) $d=4$, получим сумму d равную 14 $33\ 333\ 333\ 333 * 40 + 14 = 1\ 333\ 333\ 333\ 334$ </p>	15
4.	<pre> n = int(input()) a = [0]*n for i in range(n): a[i] = int(input()) fl = True for i in range(3,n): if (a[i-3]+a[i-2]+a[i-1]) != a[i]: fl = False break if fl: print((a[-1]+a[-2]+a[-3])+(a[-1]+a[-2]+a[-3]+a[-1]+a[-2])) else: print('NO') </pre>	20
5.	<pre> from math import sqrt N = int(input()) x = [] def dist(x, y): return sqrt((x[0]-y[0])**2+(x[1]-y[1])**2) for i in range(N): a = [int(i) for i in input().split(' ')] x.append((a[0],a[1])) Smax = 0 s = 0 for i in range(N): for j in range(N): </pre>	30

```
for m in range(N):
    if dist(x[i],x[j]) == dist(x[i],x[m]) and dist(x[i],x[j]) ==
dist((x[m][0],x[j][1]),x[m]) and dist(x[i],(x[m][0],x[j][1])) == dist(x[j],x[m])
and (dist(x[i],x[j]) * dist(x[i],x[m])>=Smax):
        Smax = dist(x[i],x[j]) * dist(x[i],x[m])
        if abs(x[m][0]+x[j][1])>s:
            s=x[m][0]+x[j][1]
if Smax==0:
    print('NO')
else: print(s)
```

Информатика. 8 класс

Критерии оценивания

1	Правильный ответ с логичным и полным объяснением	15
	Верное рассуждение с логичным и полным объяснением, ошибка округления ответа	14
	В целом верное рассуждение, ответ неверный вследствие арифметических или логических ошибок	12
	Частично верное рассуждение	10
	Верное рассуждение, ответ неверный	8
	Правильный ответ без пояснения	6
	Неверное рассуждение, неправильный ответ	3
	Другой ответ	0
2	Правильный ответ с полным объяснением	20
	Правильный ответ, объяснение не полное, нет пояснений, приведена таблица истинности	17
	Верное рассуждение, найдены не все комбинации, арифметические ошибки	15
	Частично верное рассуждение, ответ неверный	10
	Правильный ответ без пояснения	5
	Неверное рассуждение, неправильный ответ	3
	Другой ответ	0
3	Правильный ответ с полным объяснением	15
	Верное рассуждение, числовой ответ не записан	14
	В целом верное рассуждение, ответ неверный вследствие арифметических или логических ошибок	12
	Некоторые верные рассуждение, ответ неверный	9
	Правильный ответ без пояснения	5
	Неверное рассуждение, неправильный ответ	3
	Другой ответ	0
4	Правильно решающий задачу, работающий программный код на все тестах	20
	Правильно решающий задачу, работающий программный код, но неправильно проходит некоторые тесты.	18
	Работающий программный код, но есть ошибки в работе алгоритма при прохождении всех тестов	16
	Программный код работает, но есть незначительные ошибки в работе алгоритма (частично верный код)	13
	Программный код частично верный, но не работающий	10
	Программный код работающий, но большая часть алгоритма ошибочна	8
	Программный код работающий, но полностью ошибочный	5
	Описан алгоритм работы программы, но не написан программный код	2

	Другой ответ	0
5	Правильно решающий задачу, работающий на все тестах программный код	30
	Правильно решающий задачу, работающий программный код, но неправильно проходит некоторые тесты.	27
	Работающий программный код, но есть незначительные ошибки в работе алгоритма	25
	Программный код частично верный, но не работающий	20
	Программный код работающий, но большая часть алгоритма ошибочна	15
	Программный код работающий, но полностью ошибочный	10
	Описан алгоритм работы программы, но не написан программный код	6
	Другой ответ	0