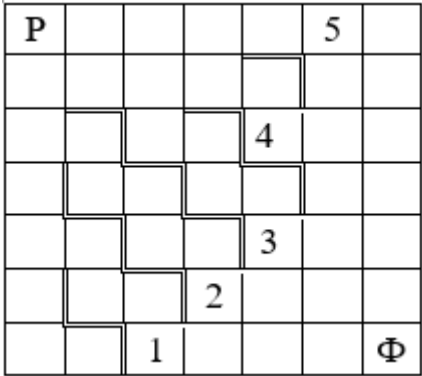


| | | | |
|----|--|----|--|
| | <p>стоит помнить, что если у нас значение энергии в ячейках сверху и слева было 1 или меньше, походить мы не можем. В таком случае в ячейку мы запишем -999, это будет означать, что мы туда не доберёмся. В P2 запишем формулу =ЕСЛИ(МАКС(P1;O2)-2>=0; МАКС(P1;O2)-2+B2; -999) и скопируем эту формулу во все ячейки диапазона P2:AA13. Таким образом, мы увидим ячейки, куда можно попасть, и до самых дальних идти ровно 20 ходов.</p> | | |
| 5. | <pre>a = int(input()) arr = list(map(int, input().split())) max_count = 0 for i in arr: counter = 0 while i: if i % 2: counter += 1 i //= 2 max_count = max(max_count, counter)</pre> <p>Пример программы, справляющейся со всеми тестами на Python.</p> | 18 | |
| 6. | <pre>N = int(input()) a = list(map(int, input().split())) bigF = [0 for i in range(101)] def f(x): if x==2: return a[1]-a[0] if x==3: return a[2]-a[0] if bigF[x]!=0: return bigF[x] bigF[x] = min(f(x-1) + a[x-1]-a[x-2], f(x-2)+a[x-1]-a[x-2]) return bigF[x] print(f(N))</pre> <p>Пример программы, справляющейся со всеми тестами на языке</p> | 25 | |

| A | B | C | $B \rightarrow C$ | $\neg A \& B$ | $A \text{ xor } C$ | $F(A, B, C)$ |
|---|---|---|-------------------|---------------|--------------------|--------------|
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 |

Информатика. 9 класс
Решения и ответы

2 вариант

| № | Правильный ответ | Балл | Прим |
|----|--|------|------|
| 1. | <p>Построим таблицы истинности для всех значений А, В, С, а также всем нужных нам функций от них. Получим, что при 4 комбинациях F(A, B, C) принимает ложное значение. Функции можно подобрать разные, самая простая (A xor not C) xor (not A and B).</p> | 12 | |
| 2. |  <p>Обратим внимание, что добраться до финиша можно через одну из обозначенных на карте клеток. Способ добраться до финиша через клетку 1 один. Затем до клетки 2 можно добраться одним способом, а от неё до клетки Ф ровно четырьмя. До клетки 3 три способа дойти и способов добраться от неё до клетки Ф, итого 18 способов. До клетки 4 четыре способа добраться и пять способов от неё до Ф, итого 20 способов. До клетки 5 можно добраться только по верху, и от неё идёт 7 возможных путей. Итого $1+4+18+20+7 = 50$</p> | 15 | |
| 3. | <p>ввести(x) напечатать($x+3*((x*x)\%3) - x\%3$)</p> | 15 | |
| 4. | <p>Ответ: 21. Для этого найдём максимальное количество ресурсов, которое может остаться у робота в каждой ячейке для таблицы. Для каждой ячейки верхней строки это будет сумма количества ресурсов у робота в левой ячейке, число ресурсов в нынешней ячейке и минус 2. Для каждой ячейки левого столбца это будет сумма всех ячеек сверху от текущей. В ячейку O1 запишем 20, в ячейку справа запишем формулу = O1+B1-2. Скопируем эту формулу во все ячейки в диапазоне P1:AA1. В ячейку O2 запишем формулу =O1+A2-2 и скопируем её вниз до O13. Для остальных ячеек будем сравнивать значение ячейки слева и значение ячейки сверху и присваивать текущей ячейке значение суммы той ячейки,</p> | 15 | |

| | | | |
|----|---|----|--|
| | <p>в которой значение больше, и текущей ячейки, минус 2. Однако стоит помнить, что если у нас значение энергии в ячейках сверху и слева было 1 или меньше, походить мы не можем. В таком случае в ячейку мы запишем -999, это будет означать, что мы туда не доберёмся. В P2 запишем формулу =ЕСЛИ(МАКС(P1;O2)-4>=0; МАКС(P1;O2)-4+B2; -999) и скопируем эту формулу во все ячейки диапазона P2:AA13. Таким образом, мы увидим ячейки, куда можно попасть, и до самых дальних идти ровно 21 ход.</p> | | |
| 5. | <pre>a = int(input()) arr = list(map(int, input().split())) min_count = 0 for i in arr: counter = 0 while i: if i % 2: counter += 1 i //= 2 min_count = min(min_count, counter) i</pre> <p>Пример программы, справляющейся со всеми тестами на Python.</p> | 18 | |
| 6. | <pre>N = int(input()) a = list(map(int, input().split())) bigF = [0 for i in range(101)] def f(x): if x==2: return a[1]-a[0] if x==3: return a[2]-a[0] if bigF[x]!=0: return bigF[x] bigF[x] = min(f(x-1) + a[x-1]-a[x-2], f(x-2)+a[x-1]-a[x-2]) return bigF[x] print(f(N))</pre> <p>Пример программы, справляющейся со всеми тестами на языке</p> | 25 | |

| A | B | C | A → C | B xor C | A & not B | F(A, B, C) |
|---|---|---|-------|---------|-----------|------------|
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 |

Информатика. 9 класс

Решения и ответы

3 вариант

| № | Правильный ответ | Балл | Прим | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----|---|------|------|--|---|---|--|---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|---|--|--|--|----|--|
| 1. | <p>По первому выражению видно, что если F от одного выражения с одной единицей и двумя нулями будет равно 0 (или 1), тогда и все остальные выражения с одной единицей должны давать 0 (или 1). Такая же ситуация для выражений с двумя единицами. Из третьего можно понять, что $F(1, 1, 1) = F(0, 0, 0) = 0$.</p> <p>Подставляем эти значения во второе и получаем $F(1, 1, 1) \vee F(0, 0, 1) = 1$, откуда следует, что $F(0, 0, 1) = 1$, значит, $F(0, 1, 0)$ и $F(1, 0, 0)$ равны 1. Если подставим во второе получаем $F(1, 1, 0) \vee F(0, 0, 0) = 1$, то имеем, что $F(1, 1, 0) = 1$, а значит, и $F(0, 1, 1)$ и $F(1, 0, 1)$ равны 1. Функцию полностью определили, задать её можно, например, как $F(A, B, C) = (A \vee B \vee C) \wedge (\neg A \vee \neg B \vee \neg C)$</p> | 12 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2. | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Р</td> <td></td> <td></td> <td>2</td> <td></td> <td></td> <td>Ф</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td>1</td> <td></td> <td></td> <td></td> </tr> </table> <p>Легко заметить, что до клетки Ф можно добраться либо по верху, условно говоря через клетку 2, и по низу, через клетку 1. Нам обязательно надо будет сделать 6 ходов вправо в обоих случаях, затем, чтобы добраться до клетки 1 надо 4 шага вниз, а потом 4 шага вверх в любом случае, итого ровно 14 единиц топлива. Такое же верно и для клетки 2. До клетки 1 можно добраться 5 способами, а оттуда до Ф 9 способами, итого 45 способов. До клетки 2 можно добраться 5 способами, а оттуда до Ф 3 способами, итого 15 способов. В сумме 60 способов.</p> | Р | | | 2 | | | Ф | | | | | | | | | | | | | | | | | | | | | | | | | 1 | | | | 15 | |
| Р | | | 2 | | | Ф | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3. | <p>вести(x) напечатать($x+5*((x*x*x*x)\%3) - x\%5$) либо ещё вариант вести(x) напечатать($(x-1)/5*5+5$)</p> | 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4. | <p>Ответ: 20.</p> <p>Для этого найдём максимальное количество ресурсов, которое может остаться у робота в каждой ячейке для таблицы. Для каждой ячейки верхней строки это будет сумма количества ресурсов у робота в левой ячейке, число ресурсов в нынешней</p> | 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | |
|----|---|----|--|
| | <p>ячейке и минус 2. Для каждой ячейки левого столбца это будет сумма всех ячеек сверху от текущей. В ячейку O1 запишем 10, в ячейку справа запишем формулу $=O1+B1-2$. Скопируем эту формулу во все ячейки в диапазоне P1:AA1. В ячейку O2 запишем формулу $=O1+A2-2$ и скопируем её вниз до O13. Для остальных ячеек будем сравнивать значение ячейки слева и значение ячейки сверху и присваивать текущей ячейке значение суммы той ячейки, в которой значение больше, и текущей ячейки, минус 2. Однако стоит помнить, что если у нас значение энергии в ячейках сверху и слева было 1 или меньше, походить мы не можем. В таком случае в ячейку мы запишем -999, это будет означать, что мы туда не доберёмся. В P2 запишем формулу $=ЕСЛИ(МАКС(P1;O2)-2>=0; МАКС(P1;O2)-2+B2; -999)$ и скопируем эту формулу во все ячейки диапазона P2:AA13. Теперь в ячейках, где стена находится слева, формулу можно заменить на $=ЕСЛИ(S2-2>=0; S2+E3-2; -999)$, а в ячейках, где стена сверху, на $=ЕСЛИ(P12-2>=0; P12+C12-2; -999)$ Таким образом, мы увидим ячейки, куда можно попасть, и до самых дальних идти ровно 20 ходов.</p> | | |
| 5. | <pre> a = int(input()) arr = list(map(int, input().split())) max_count = 0 for i in arr: counter = 0 while i: if i % 4: counter += 1 i //= 4 max_count = max(max_count, counter) </pre> <p>Пример программы, справляющейся со всеми тестами на Python.</p> | 18 | |
| 6. | <pre> N = int(input()) a = list(map(int, input().split())) bigF = [0 for i in range(101)] def f(x): if x==2: return a[1]-a[0] if x==3: return a[2]-a[0] if bigF[x]!=0: return bigF[x] bigF[x] = min(f(x-1) + a[x-1]-a[x-2], f(x-2)+a[x-1]-a[x-2]) return bigF[x] print(f(N)) </pre> <p>Пример программы, справляющейся со всеми тестами на языке</p> | 25 | |

Информатика. 9 класс

Решения и ответы

4 вариант

| № | Правильный ответ | Балл | Прим |
|----|--|------|------|
| 1. | <p>49 комбинаций</p> <p>Из второго видно, что функция на комбинациях 0,1,1,1 и 0,0,0,0 равна нулю. Если функция на таких комбинациях равна нулю, на противоположных 1 (из первого утверждения). Затем если рассматривать комбинации с 0 на первом месте и одной единичкой, то если хотя бы на одной такой комбинации функция равна 1, то на всех остальных комбинациях такого же вида она должна быть равна 0 (это видно из второго, единичка там будет будто бы прокручиваться). Если F на всех комбинациях такого рода равна 0, то на противоположных вида 1,0,1,1 и 1,1,0,1 и 1,1,1,0 F равен должен быть 1, но если где-то он всё же равен 1, это даёт нам выбор из двух вариантов ещё и для «противоположной» комбинации, итого $1+2+2+2=7$. Для комбинации с нулём в начала и двумя единичками рассуждения аналогичны, итого $7*7=49$ вариантов.</p> | 12 | |
| 2. | <p>Ответ 360, спросить надо про 9.</p> <p>Сами три буквы в любом случае можно расставить $6*5*4/6=20$ способами. Способы разбить 21 на три слагаемых $9+9+3=9+8+4=9+7+5=9+6+6=8+8+5=8+7+6=7+7+7$. Если учесть ещё и перестановки внутри, получится 28. Если спросить про 3, то если её нет, вариантов 25. Если спросить про 5, то если её нет (худший случай), придётся перебрать 19 вариантов. С 9 в худшем случае будет 18 вариантов, итого $18*20$ вариантов.</p> | 15 | |
| 3. | <p>Решений нет. Для начала заметим, что x и y должны делиться на три для того, чтобы равенство было верным. Затем y окажется, что наша сумма всегда имеет остаток 1 при делении на три при любых x и y и нулю равняться не может.</p> | 15 | |
| 4. | 9994 | 15 | |
| 5. | <p>Идея программы состоит в том, чтобы добавить в сумму все положительные чётные числа. Затем, если нечётных положительных чисел чётное число, то всех их добавим, если же нет, либо не добавим самое маленькое, либо добавим его с наименьшим отрицательным (из этих двух вариантов надо выбрать наиболее выгодный).</p> | 18 | |
| 6. | <pre>N = int(input()) import sys k2, k3, k5, k7 = 0, 0, 0, 0 if(N==1): print(1) elif(N==0):</pre> | 25 | |

| | | |
|--|--|--|
| <pre>print(10) sys.exit() while(N%2==0): k2 += 1 N/=2 while(N%3==0): k3 += 1 N/=3 while(N%5==0): k5 += 1 N/=5 while(N%7==0): k7 += 1 N/=7 if(N != 1): print(-1) else: k9 = int(k3//2) k3 = k3%2 k8 = int(k2//3) k2 = k2%3 k6 = 0 k4 = 0 if(k3 != 0 and k2 != 0): k6 = 1 k3 -= 1 k2 -= 1 if(k2 == 2): k4 = 1 k2 = 0 for i in range(k2): print(2, sep = "", end = "") for i in range(k3): print(3, sep = "", end = "") for i in range(k4): print(4, sep = "", end = "") for i in range(k5): print(5, sep = "", end = "") for i in range(k6): print(6, sep = "", end = "") for i in range(k7): print(7, sep = "", end = "") for i in range(k8): print(8, sep = "", end = "") for i in range(k9): print(9, sep = "", end = "")</pre> | | |
|--|--|--|

Информатика. 9 класс
Критерии оценивания

1. Правильное решение – 12 баллов.

За некоторые подвижки в решении могло ставиться от 5 до 10 баллов.

Неправильное решение – 0 баллов.

2. Правильное решение – 15 баллов.

Решение с небольшими недочётами – 10-12 баллов.

За какие-то правильные шаги, но в целом неправильное решение 5 баллов.

Неправильное решение – 0 баллов.

3. Правильное решение – 15 баллов.

Решение с небольшими недочётами – 10-12 баллов.

Только за идею того, что надо использовать остатки не более 5 баллов.

Неправильное решение – 0 баллов.

4. Правильное решение – 15 баллов.

Если было не учтена ситуация, что робот ушёл в минус по топливу, но сразу же его в клетке восстановил (это давало в решении лишний шаг), то снимать 5 баллов.

Неправильное решение – 0 баллов.

5. За все правильные тесты ставится 18 баллов. За ошибку в одном тесте 15 баллов, если правильный ответ дан только на половину тестов, то ставить 10 баллов.

6. За все правильные тесты ставится 25 баллов. За ошибку в одном тесте 20 баллов, если правильный ответ дан только на первый тест, то ставить 5 баллов.