B1 (базовый уровень, время – 4 мин)

Тема: Поиск алгоритма минимальной длины для исполнителя.

Что нужно знать:

- каких-либо особых знаний из курса информатики не требуется, задача решаема на уровне 6-7 класса простым перебором вариантов, просто его нужно организовать оптимальным образом
- *исполнитель* – это человек, группа людей, животное, машина или другой объект, который может понимать и выполнять некоторые команды

Пример задания:

У исполнителя Калькулятор две команды, которым присвоены номера:

```
1. прибавь 3
2. умножь на 4
```

Выполняя первую из них, Калькулятор прибавляет к числу на экране 3, а выполняя вторую, умножает его на 4. Запишите порядок команд в программе получения из числа 3 числа 57, содержащей не более 6 команд, указывая лишь номера команд.

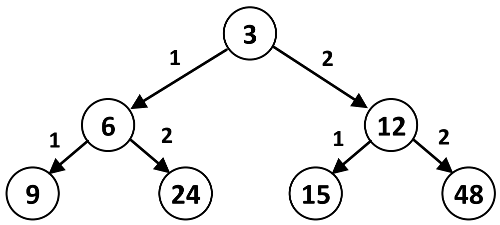
(Например, программа 21211 это программа

```
умножь на 4
прибавь 3
умножь на 4
прибавь 3
прибавь 3
```

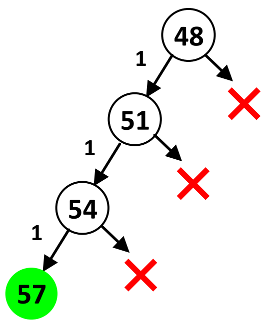
которая преобразует число 2 в 50.)

Решение (вариант 1, «прямой ход»):

1) обратим внимание, что в условии ограничено число команд, поэтому неявно ставится задача написать самую короткую программу для решения задачи
2) начнем решать задачу, «отталкиваясь» от начального числа
3) на первом шаге с помощью имеющихся команд из числа 3 можно получить 6 или 12;
4) на втором шаге из 6 можно получить 9 и 24, а из 12 – 15 и 48, и т.д., получается такая схема (структура «дерево»), цифры около стрелок показывает номер выполненной команды:



5) уже чувствуется, что дерево сильно разрастается, на следующем уровне будет уже 8 вариантов, потом – 16 и т.д. (на каждом следующем уровне – в 2 раза большем, чем на предыдущем)
6) нужно выбрать такой план дальнейшего перебора вариантов, который может быстрее всего привести к цели (числу 57)
7) видим, что после второй операции ближе всего к результату оказалось число 48, попробуем начать анализ с этой ветки; если не получится – возьмем число 24 и т.д.
8) ветка дерева, начиная от числа 48, построена на рисунке справа; красный крестик показывает, что полученное значение превышает 57



9) итак, мы вышли на число 57 в результате такой последовательности команд: 22111, ее длина равна 5, что удовлетворяет условию задачи.
10) таким образом, правильный ответ – 22111.

Возможные ловушки и проблемы:
- большую схему неудобно рисовать, в ней легко запутаться
- не всегда можно сразу угадать нужную ветку «дерева», то есть, ту, которая быстрее всего приведет к успеху

Решение (вариант 2, «обратный ход»):
1) нам нужно увеличить число (с 3 до 57), для этого в большинстве случаев умножение эффективнее сложения, поэтому нужно постараться максимально использовать умножение, а сложение – только в крайних случаях
2) попробуем решить задачу «обратным ходом», начав с числа 57;
3) очевидно, что последней командой не может быть умножение на 4 (57 на 4 не делится), поэтому последняя команда – сложение (**`прибавь 3`**), над стрелкой записан номер команды:

$$...54 \xrightarrow{1} 57$$

4) число 54 также не делится на 4, поэтому предыдущая команда – тоже сложение:

$$...51 \xrightarrow{1} 54 \xrightarrow{1} 57$$

5) аналогично для числа 51:

$$...48 \xrightarrow{1} 51 \xrightarrow{1} 54 \xrightarrow{1} 57$$

6) число 48 делится на 4, поэтому используем умножение:

$$...12 \xrightarrow{2} 48 \xrightarrow{1} 51 \xrightarrow{1} 54 \xrightarrow{1} 57$$

7) наконец, добавив в начало программы еще одно умножение, получаем полную цепочку:

$$3 \xrightarrow{2} 12 \xrightarrow{2} 48 \xrightarrow{1} 51 \xrightarrow{1} 54 \xrightarrow{1} 57$$

8) таким образом, правильный ответ – 22111, эта программа состоит из 5 команд.

Возможные ловушки и проблемы:
- иногда может потребоваться «откат» назад, например, если исходное число – 6, то применив деление на 4 для 12 мы «проскакиваем» его (получаем `12/4=3<6`), поэтому нужно возвращаться обратно к 12 и дважды применять сложение; в этом случае ответ будет такой:

$$6 \xrightarrow{1} 9 \xrightarrow{1} 12 \xrightarrow{2} 48 \xrightarrow{1} 51 \xrightarrow{1} 54 \xrightarrow{1} 57$$

Почему здесь «обратный ход» лучше?:
- обратим внимание, что когда мы «шли» в обратном направлении, от конечного числа к начальному, часто очередную операцию удавалось определить однозначно (когда число не делилось на 4)
- это связано с тем, что среди допустимых команд есть «не всегда обратимая» операция – умножение: умножить целое число на 4 можно всегда, а разделить нацело – нет; в подобных случаях результат быстрее получается именно «обратным ходом», во время которого сразу отбрасываются невозможные варианты

Еще пример задания:

У исполнителя, который работает с положительными однобайтовыми двоичными числами, две команды, которым присвоены номера:

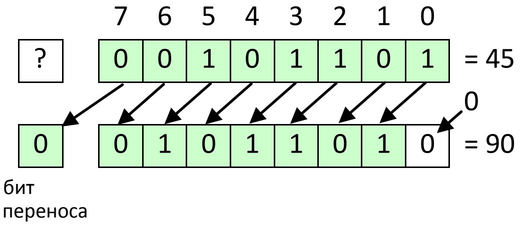
`1. сдвинь влево`

`2. вычти 1`

Выполняя первую из них, исполнитель сдвигает число на один двоичный разряд влево, а выполняя вторую, вычитает из него 1. Исполнитель начал вычисления с числа 104 и выполнил цепочку команд 11221. Запишите результат в десятичной системе.

Решение:

1) важно, что числа однобайтовые – на число отводится 1 байт или 8 бит
2) главная проблема в этой задаче – разобраться, что такое «сдвиг влево»; так называется операция, при которой все биты числа в ячейке (регистре) сдвигаются на 1 бит влево, в младший бит записывается нуль, а старший бит попадает в специальную ячейку – *бит переноса*:



можно доказать, что в большинстве случаев результат этой операции – умножение числа на 2, однако есть исключение: если в старшем (7-ом) бите исходного числа x была 1, она будет «выдавлена» в бит переноса, то есть потеряна[1], поэтому мы получим остаток от деления числа $2x$ на 2^8=256

3) попутно заметим, что при сдвиге вправо[2] в старший бит записывается 0, а младший «уходит» в бит переноса; это равносильно делению на 2 и отбрасыванию остатка
4) таким образом, фактически команда **`сдвинь влево`** означает **`умножь на 2`**
5) поэтому последовательность команд 11221 выполняется следующим образом

Код команды	Действие	Результат	Примечание
		104	
1	**`умножь на 2`**	208	
1	**`умножь на 2`**	160	остаток от деления 208*2 на 256
2	**`вычти 1`**	159	
2	**`вычти 1`**	158	
1	**`умножь на 2`**	60	остаток от деления 158*2 на 256

6) правильный ответ – 60.

Еще пример задания:

Исполнитель Робот действует на клетчатой доске, между соседними клетками которой могут стоять стены. Робот передвигается по клеткам доски и может выполнять команды 1 (вверх), 2 (вниз), 3 (вправо) и 4 (влево), переходя на соседнюю клетку в направлении, указанном в

[1] Используя ассемблер (язык машинных кодов с символьными командами), можно добраться до бита переноса и использовать его.

[2] Кроме *логического* сдвига вправо, о котором идет речь, есть еще *арифметический*, при котором старший бит не меняется.

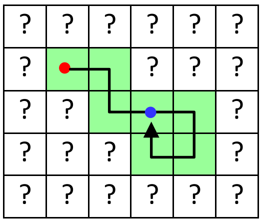
скобках. Если в этом направлении между клетками стоит стена, то Робот разрушается. Робот успешно выполнил программу

```
3233241
```

Какую последовательность из трех команд должен выполнить Робот, чтобы вернуться в ту клетку, где он был перед началом выполнения программы, и не разрушиться вне зависимости от того, какие стены стоят на поле?

Решение:

1) фактически заданная программа движения Робота, которую он успешно выполнил, показывает нам свободный путь, на котором стенок нет
2) поэтому для того, чтобы не разрушиться на обратном пути, Робот должен идти точно по тому же пути в обратном направлении
3) нарисуем путь Робота, который выполнил программу 3233241:



Робот начал движение из клетки, отмеченной красной точкой, и закончил в клетке, где стоит синяя точка

4) чтобы вернуться в исходную клетку (с красной точкой) по пройденному пути, Роботу нужно сделать шаг влево (команда 4), затем шаг вверх (команда 1) и еще один шаг влево (команда 4)
5) таким образом, ответ – 414.

Еще пример задания:

Исполнитель Робот ходит по клеткам бесконечной вертикальной клетчатой доски, переходя по одной из команд вверх, вниз, вправо, влево в соседнюю клетку в указанном направлении. Робот выполнил следующую программу:

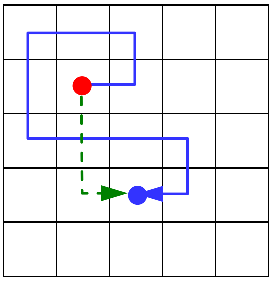
```
вправо
вверх
влево
влево
вниз
вниз
вправо
вправо
вправо
вниз
влево
```

Укажите наименьшее возможное число команд в программе, переводящей Робота из той же начальной клетки в ту же конечную.

Решение (способ 1, моделирование движения Робота):

1) отметим, что в условии ничего не говорится о стенках, то есть, молчаливо предполагаем, что их нет

2) можно повторить все движения Робота на бумажке и посмотреть, куда он уйдет; на схеме исходная точка обозначена красной точкой, а конечная – синей, синяя линия показывает путь Робота:



3) поскольку Робот не может ходить по диагонали, для перехода из начальной точки в конечную кратчайшим путем ему нужно выполнить, например, такую программу (см. штриховые линии на рисунке):

```
вниз
вниз
вправо
```

4) есть и другие варианты (попробуйте их найти!), но все они содержат 3 команды: одну команду **вправо** и две команды **вниз**
5) таким образом, ответ – 3.

Решение (способ 2, анализ программы):

1) можно решить задачу без повторения движений Робота
2) обратим внимание, что пары команд «вперед-назад» и «влево-вправо» дают нулевой эффект, то есть, не перемещают Робота, поэтому все такие пары можно выкинуть из программы
3) поскольку стенок нет, все равно где стоят парные команды в программе, вычеркиваем их:

~~**вправо**~~
~~**вверх**~~
~~**влево**~~
~~**влево**~~
~~**вниз**~~
вниз
~~**вправо**~~
вправо
~~**вправо**~~
вниз
~~**влево**~~

4) смотрим, какие команды остались (они отмечены желтым маркером), их всего 3
5) таким образом, ответ – 3.

Еще пример задания:

Исполнитель КУЗНЕЧИК живёт на числовой оси. Начальное положение КУЗНЕЧИКА – точка 0. Система команд Кузнечика:

Вперед 4 – *Кузнечик прыгает вперед на 4 единицы,*

Назад 3 – *Кузнечик прыгает назад на 3 единицы.*

Какое наименьшее количество раз должна встретиться в программе команда «Назад 3», чтобы Кузнечик оказался в точке 27?

Решение (составление уравнения, подбор решения):

1) обозначим через x количество команд «Вперед 4» в программе, а через y – количество команд «Назад 3»
2) для того, чтобы КУЗНЕЧИК попал в точку 27 из точки 0, должно выполняться условие
$$4x-3y=27-0=27$$
3) это уравнение называется *диофантовым*; поскольку числа 4 и 3 – взамнопростые (их наибольший общий делитель равен 1), оно имеет бесконечно много решений
4) из всех решений нас интересует такое, при котором y – наименьшее возможное неотрицательное (!) число
5) представим уравнение в виде
$$4x=27+3y$$
нужно подобрать минимальное неотрицательное y, при котором правая часть делится на 4
6) дальше используем метод подбора (или перебора), начиная от 1; получаем
$y=0 \quad\Rightarrow\quad 4x=27$
$y=1 \quad\Rightarrow\quad 4x=30$
$y=2 \quad\Rightarrow\quad 4x=33$
$y=3 \quad\Rightarrow\quad 4x=36$
...
7) видим, что первое y, при котором $27+3y$ делится на 4, это $y=3$ (при этом $x=9$).
8) таким образом, ответ – 3.