

Решения

Вариант 1

Задача 1.

Составляем уравнение $3k+7+2k+1=6k$. Отсюда $k=8$. Можно просто угадать основание, но в этом случае должна быть сделана проверка.

Задача 2.

Вначале нужно «пропрыгать» вправо до конца, а потом возвращаться прыжками и закрашивать ячейки.

Повтори 6 раз Вперед 3

Повтори 8 раз (Закрась, Назад 2)

Задача 3.

Для перебора всех возможных вариантов можно составить дерево или таблицу

1-я цифра	2-я цифра	3-я цифра	4-я цифра	5-я цифра	6-я цифра
1 (+ - - + -)	11 (- - + -)	112 (- + -)	1122 (+ -)	11221 (-)	112212
				11225	
		114 (+ -)	1141 (-)	11412	
		1145			
	15 (- + -)	152 (+ -)	1521 (-)	15212	
			1525		
3 (- - + -)	32 (- + -)	322 (+ -)	3221 (-)	32212	
			3225		
	34 (+ -)	341 (-)	3412		
		345			

В скобках указана оставшаяся последовательность

Всего 10 вариантов дешифровки, возможные варианты выделены в таблице жирным шрифтом.

Задача 4.

Всего возможных паролей (без учета подсказок) $16^4=65536$ (всего 4 места для символов, всего различных символов 16))

С учетом первой подсказки: $4*16^3=16384$ (буква 'А' может стоять на первом, втором, третьем или четвертом первом месте, остальные символы любые (всего разных символов 16))

С учетом второй подсказки: $15^4= 50625$ (всего 4 места для символов, всего различных символов 16))

С учетом третьей подсказки: $4*10*6^3= 8640$ (цифр всего 10, цифра может стоять на любом месте, остальные символы – буквы, их всего 6)

Поэтому Бельчонку следует использовать третью подсказку.

Задача 5.

Возможно применение любого алгоритма, приводящего к верному решению.
Пример программы на языке Turbo Pascal:

```
var
  store: integer; //количество шишек в наличии у Бельчонка (запас)
  N: integer; //количество ведер
  k,dist,count,result,distantion: integer;
begin
  ReadLn(N);
  store := 5; //начальный запас
  distantion := 0; //начальная позиция Бельчонка
  result := -1;
  for k:=1 to N do begin
    ReadLn(dist); //считываем расстояние от точки старта до ведра
    ReadLn(count); //считываем количество шишек в ведре
    If store - (dist-distantion) < 0 then begin //доехать он не смог, не хватило
запаса
      result := distantion + store; //вычисляем сколько он реально смог проехать и
выходим из цикла
      break;
    end
    else begin
      store := store-(dist-distantion)+count; //вычисляем новый запас у учетом
сколько съел Бельчонок и содержимого ведра
      If store > 10 then store := 10;
      distantion := dist;
    end;
  end;
  If result = -1 then result := distantion+store;
  WriteLn(result);
  ReadLn;
end.
```

Вариант 2

Задача 1.

Составляем уравнение $3k+3+2k+5=6k+1$. Отсюда $k=7$. Можно просто угадать основание, но в этом случае должна быть сделана проверка.

Задача 2.

Вначале нужно «пропрыгать» влево до конца, а потом возвращаться прыжками и закрашивать ячейки.

Повтори 11 раз Назад 2

Повтори 6 раз (Вперед 3, Закрась)

Задача 3.

Для перебора всех возможных вариантов можно составить дерево или таблицу

1-я цифра	2-я цифра	3-я цифра	4-я цифра	5-я цифра	6-я цифра
1 (+ + + - +)	12 (+ + - +)	122 (+ - +)	1222 (- +)	12221 (+)	122212
				12225	
		123 (- +)	1231 (+)	12312	
			1235		
	13 (+ - +)	132 (- +)	1321 (+)	13212	
			1325		
5 (+ + - +)	52 (+ - +)	522 (- +)	5221 (+)	52212	
			5225		
	53 (- +)	531 (+)	5312		
		535			

В скобках указана оставшаяся последовательность

Всего 10 вариантов дешифровки, возможные варианты выделены в таблице жирным шрифтом.

Задача 4.

Всего возможных паролей (без учета подсказок) $16^4=65536$ (всего 4 места для символов, всего различных символов 16))

С учетом первой подсказки: $4*16^3=16384$ (буква 'F' может стоять на первом, втором, третьем или четвертом первом месте, остальные символы любые (всего разных символов 16))

С учетом второй подсказки: $4*15^3= 13500$ (всего 4 места для нуля, остальные символы любые, всего различных символов 15)

С учетом третьей подсказки: $4*6*10^3= 24000$ (букв всего 6, буква может стоять на любом месте, остальные символы – цифры, их всего 10)

Поэтому Бельчонку следует использовать вторую подсказку.

Задача 5.

Возможно применение любого алгоритма, приводящего к верному решению.

Пример программы на языке Turbo Pascal:

```
var
  store: integer; //количество шишек в наличии у Бельчонка (запас)
  N: integer; //количество ведер
  k,dist,count,result,distantion: integer;
begin
  ReadLn(N);
  store := 3; //начальный запас
  distantion := 0; //начальная позиция Бельчонка
  result :=-1;
  for k:=1 to N do begin
    ReadLn(dist); //считываем расстояние от точки старта до ведра
```

```
ReadLn(count); //считываем количество шишек в ведре
If store - (dist-distantion) < 0 then begin //доехать он не смог, не хватило
запаса
  result := distantion + store; //вычисляем сколько он реально смог проехать и
выходим из цикла
  break;
end
else begin
  store := store-(dist-distantion)+count; //вычисляем новый запас у учетом
сколько съел Бельчонок и содержимого ведра
  If store > 5 then store := 5;
  distantion := dist;
end;
end;
If result = -1 then result := distantion+store;
WriteLn(result);
ReadLn;
end.
```