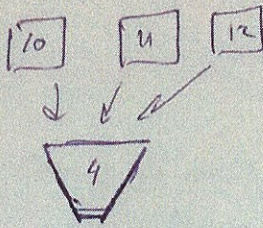


НУО: } в 4-ом стакане есть следы реактива.



Тогда используя 2 полоски можно понять, в какой именно канистре (10, 11 или 12) есть реактив.

В таком образом, мы всегда можем определить где находится заданный реактив используя всего лишь (3+2) полоски, а если их шесть, то тем более.

или

будем писать на Java:

import java.util.Scanner;

public class game {

public static void main (String[] args) {

int [][] desk = new int [8][8];
int distance; //i

while (true) {

String str = in.nextLine();

switch (str) {

case "show": for (int i=0; i<8; i++) {
for (int j=0; j<8; j++)
System.out.print(desk[i][j] + " ");
System.out.println();
} break;

case "distance": System.out.print("Введите расстояние
перемещения фигурок
за одно действие");

double distance = (int) in.nextInt() / 12.5;

System.out.print("Перемещение за одно
действие осуществляется на 1px");

break;

}

ω1

6 известв $\Rightarrow$ 3 бита НАПРЯКУ

Нарисунков: $\frac{230 \times 470.3}{8}$ байт

НА СТРАНИЦУ ТЕКСТА: $\frac{460 \cdot 16}{2} = 920$ байт

Вариант 3

$$2,5 \text{ Мбайт} = 2,5 \cdot 2^{20} \text{ байт}$$

3

n - кол-во страниц текста, k - кол-во рисунков

$$\begin{cases} n \cdot 920 \cdot 80 = \frac{23 \cdot 47 \cdot 75}{2} k \Rightarrow n = \frac{47}{64} k \\ n \cdot 920 + \frac{23 \cdot 47 \cdot 75}{2} k = 2,5 \cdot 2^{20} \end{cases}$$

$$\frac{47}{64} k \cdot 920 + \frac{23 \cdot 47 \cdot 75}{2} k = 2,5 \cdot 2^{20}$$

$$k = \frac{2,5 \cdot 2^{20}}{23 \cdot 47 (\frac{5}{8} + 37,5)}$$

$$n = \frac{47}{64} k = \frac{2,5 \cdot 2^{20}}{23 \cdot 23 (\frac{5}{8} + 37,5)}$$

Ответ: $n+k = \frac{2,5 \cdot 2^{20}}{23 \cdot 47 (\frac{5}{8} + 37,5)} (1 + \frac{47}{64})$

ω4 Будем k в городе подписывать кратчайшее расстояние $go A$ и из какого предположено города нужно выйти, чтобы его достичь.

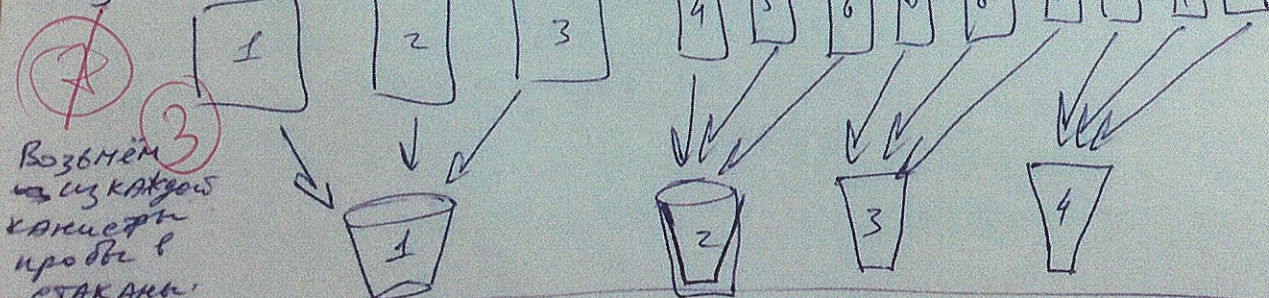
город	A	B	C	D	F	E	G	H	J	I	K
расстояние		1	2	1	4	6	6	8	9	10	12
откуда		A	B	A	D	F	C	E	G	H	I

Восстановим путь

ADFEH_IK

Ответ: ADFEHIK

ω5 Канистры:



Возьмём из каждой канистры по пробе в стаканы! Согласно схеме

За 3 полоски не сложно определить в каком стакане есть реактив (проверим любые три, если нигде нет следов, то в 4-ом, иначе там где они есть)

3/4

-19 БУДЕМ ИЩЕМ НА JAVA:

ВАРИАНТ 3

```

import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int String command = in.nextLine(); int stop int[] a = new int[128] = new stop int[128];
        switch (command) {
            case "cycles": System.out.println("Введите номер ящика");
                int i = in.nextInt();
                a[i] System.out.println("Ящик" + i + " был  
перезаписан." + a[i*4] + " раз"); break;
            case "add": System.out.println("Введите время измерения  
в часах");
                int i = 0; i += 4;
                for (int i = 0; i < 128; i += 4)
                    if (a[i-4] > a[i])
                        break;
                if (i > 125) i = 0;
                a[i]++; a[i+1] = in.nextInt();
                System.out.println("Введите время измерения в  
минутах");
                a[i+2] = in.nextInt();
                System.out.print("Время измерения " + a[i+1] +
                    " " + a[i+2] + ". Введите значение температуры  
в градусах Цельсия");
                a[i+3] = in.nextInt() + 50;
                System.out.println("Добавлено измерение: " +
                    a[i+1] + " " + a[i+2] + " " + a[i+3] + " " + (a[i+3]-50));
            case "all": for (int i = 0; i < 128; i += 4)
                System.out.println(a[i+1] + " " + a[i+2] + " " + a[i+3] + " " + (a[i+3]-50));
                break;
            case "test": System.out.print("В энергонезависимую память  
возможно записать."); int s = 0;
                int n = for (int i = 0; i < 128; i += 4)
                    s += a[i];
                System.out.println((8000 - s) + " измерений");
        }
    }
}

```

}}}

Заметка: перво я подумал, что в последних двух ящиках можно хранить текущую позицию ~~и~~ записи и кол-во циклов перезаписи. Так больше можно будет хранить ~~в~~ измерений ~~и~~ ~~каждое~~ ~~время~~ ~~цикла~~ ~~перезаписи~~ кол-во циклов перезаписи ящика, которая ~~но~~ ~~реже~~ ~~даже~~ хранит ~~также~~ текущую позицию всего лишь 250, что не позволяет увеличить циклов перезаписи всей программы.

ш 6

возможные разрешённые комбинации

Вариант 3

1 РАЗРЕШЕНО 2 1

* - горит

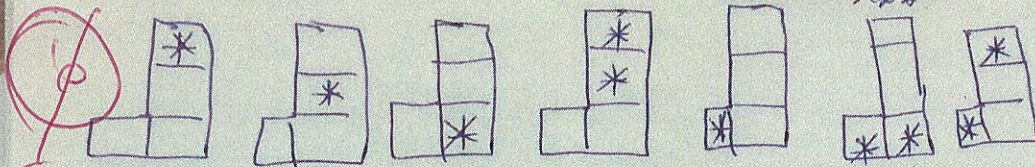


таблица истинности

к	ж	з	дз	разрешённое состояние
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0
А	Б	В	Г	

ДНФ:

$$\bar{k} \wedge \bar{j} \wedge \bar{z} \wedge \bar{dz} \vee \bar{k} \wedge \bar{j} \wedge \bar{z} \wedge dz \vee$$

$$\vee \bar{k} \wedge \bar{j} \wedge z \wedge \bar{dz} \vee \bar{k} \wedge \bar{j} \wedge z \wedge dz \vee$$

$$\vee k \wedge \bar{j} \wedge \bar{z} \wedge \bar{dz} \vee k \wedge \bar{j} \wedge \bar{z} \wedge dz \vee$$

$$\vee k \wedge \bar{j} \wedge z \wedge \bar{dz} \vee$$

$$= \bar{k} (\bar{j} + z + dz) \vee k \bar{j} (z \vee dz) \vee (k \vee j) \wedge (\bar{z} \wedge \bar{dz}) \vee$$

$$\vee k \wedge \bar{j} \wedge \bar{z} \wedge dz \vee$$

$$= \bar{k} \vee \bar{j} (\bar{k} \wedge (z \vee dz) \vee k \wedge \bar{z} \wedge dz) \vee (k \vee j) \wedge (\bar{z} \wedge \bar{dz}) \vee$$

$$= \bar{k} \vee \bar{j}$$

ш 7 $N=60$, по условию

после 1-ого цикла

$$i=60, t=0, M=30$$

на первом

на каждой ветке операции второго цикла:

$$A[i-1] \geq M \Leftrightarrow 59 \geq 30, \text{ верно}$$

выполняется вложенный цикл:

после будут поменяться два последних

элемента из A с конца послед.на каждом шаге t увеличивается на 1.

$$\text{В итоге } A[1]=60;$$

$$A[2]=1, A[3]=2, \dots, A[60]=59$$

$$t=59$$

на втором: $t=58$, и так далеепока $A[60] > 30$, то есть 30 раз выполняется второй цикл

$$t = 59 + 58 + 57 + \dots + 30 = \frac{30+59}{2} \cdot 30 = 89 \cdot 15 = 890 + 445 = 1335$$

Ответ: результат алгоритма. Вывод числа 1335.