

Работа призера заключительного этапа
командной инженерной олимпиады школьников
Олимпиада Национальной технологической инициативы

Профиль «ИНТЕЛЛЕКТУАЛЬНЫЕ ЭНЕРГЕТИЧЕСКИЕ СИСТЕМЫ»

Хайруллин Мурат Ильгизович

Класс: 9

Город: Уфа

Школа: МБОУ “Гимназия №3”

**Регион: Республика
Башкортостан**

Уникальный номер участника: 89

**Команда на заключительном
этапе: Тесла**

Результаты заключительного этапа:

№	Индивидуальный этап										Командный этап				ИТОГ
	Физика				Математика				За задачи	БАЛЛ в %	Очки	Балл	Итоговый	Командный	
89	4	0	0	0	14	0	10	0	28	11	7610,2	135,1	155	60	71

Индивидуальная часть

Персональный лист участника с номером 89:



Олимпиада НТИ

ФИО Хайруллин Мурат Ильгизович

Город Уфа

Школа № Гимназия № 3

Командная инженерная олимпиада «Олимпиада НТИ»

Направление Интеллектуальные Энергетические Системы

Предмет Математика

Номер участника 89

1/2/3/4/5/6
14/0/10/0/24

№1
 $x - xy + y = 1 \Rightarrow 1 + xy = x + y$ (Произведение чисел должно быть на 1 меньше, чем их сумма.)

Подобное условие выполняется, ^{тогда} когда один из слагаемых равен "1" ("1" не подходит т.к. ответ отрицательный)

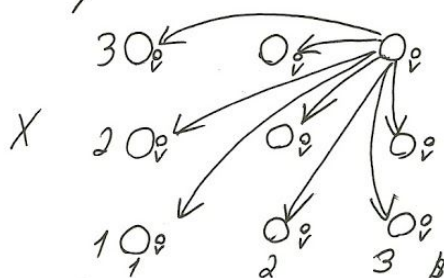
Пример: $x=6; y=-1; 6-6-1=-1$

$\Rightarrow$ одна из переменных обязательно должна равняться единице.

Ответ: $x=\text{любое}; y=1$; или $x=1; y=\text{любое}$

№2
Нет, не может т.к. на одно место в проекте претендует один кандидат, а так же количество проектов и мест ограничено. $\Rightarrow$ Будет существовать самый интересный и самый высокооплачиваемый проект, при переходе из которого не будет выполняться условие о том, что каждый новый проект должен быть или интереснее или крупнее

Пример:



○ - проекты
♀ - люди
→ - переходы
X - интересность
H - прибыльность
1-2-3 = степени интересности / прибыльности

$$\begin{array}{r} 11110 \text{ ①} \\ \underline{11111} \\ 1 \text{ ② ①} \\ \underline{11111} \end{array}$$

$$\begin{array}{r} 33331 \text{ ②} \\ \underline{33334} \\ 3 \text{ ②} \\ \underline{33334} \end{array}$$

$$\begin{array}{r} 44441 \text{ ③} \\ \underline{44445} \\ 4 \text{ ③} \\ \underline{44445} \end{array}$$

то, сколько раз
не хватает до единицы

$$\begin{array}{r} 12 \\ \hline 133332 \end{array} \textcircled{1} \quad \begin{array}{r} 12 \\ \hline 133336 \end{array} \textcircled{2} \quad \begin{array}{r} 12 \\ \hline 133335 \end{array} \textcircled{3} \Rightarrow$$

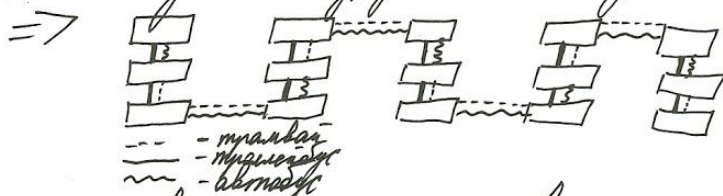
$$\Rightarrow \begin{array}{r} 12 \\ \hline 133332 \end{array} \textcircled{1} > \begin{array}{r} 12 \\ \hline 133335 \end{array} \textcircled{3} > \begin{array}{r} 12 \\ \hline 133336 \end{array} \textcircled{2} \text{ m.k. } \text{amo}$$

остатки $\Rightarrow$ больше та дробь, у которой остаток меньше $\Rightarrow$

$$\Rightarrow \frac{11110}{11111} < \frac{44441}{44445} < \frac{33331}{33334}$$

Amber: $\frac{11110}{11111}$; $\frac{44441}{44445}$; $\frac{33331}{33334}$

Исходя из условия, каждая остановка должна быть связана с другой минимум 2 типами связи $\Rightarrow$



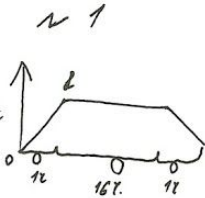
Отвѣт: 28 маршутав

Командная инженерная олимпиада «Олимпиада НТИ»

Направление Интеллектуальные Энергетические Системы

Предмет Физика

Номер участника 89



$$I = 136 \text{ A} \cdot \text{Вм} / \text{м}^2$$

$$136 \text{ A} \cdot 300 \text{ м}^2 = 40800 \text{ Вм} - \text{за } 16 \text{ ч.}$$

на т.к. отражается 99%, а поглощается

$$90\% \Rightarrow 365399,1 - \text{за } 16 \text{ ч.}$$

$$Q = cm \Delta t \quad +4\% \quad m = \frac{Q}{c \Delta t} ; \quad m = \frac{365399,1}{4000 \frac{\text{Дж}}{\text{кг} \cdot \text{C}} \cdot (590 - 20)} =$$

$$= \frac{365399,1}{120000} = 0,3044 \text{ кг} - \text{за } 16 \text{ часов}$$

Ответ: 0,3044 кг

~ 2

1	2	3	4	Σ
4	0	0	0	4

(40) за
предположение о
нагревании
теплоемкости и
заданные данные
или какая
теплота

Результаты выступления команды «Тесла»



Команда «Тесла» за терминалом финального задания.



Терминал финального задания команды «Тесла»



Задание «Импульсный повышающий преобразователь»

Результат: КПД 86,4%, что приводит к КПД солнечных батарей в игре в 115,6%.

Uвх	Iвх	Uвых	Rнаг	P1	P2	КПД
7,91	1,026	23,98	82	8,11566	7,012688	0,864093

Основное командное задание

Результат: 7610,2 очков, или 60 баллов.

Конфигурация сети:

- Потребитель 3 класса с ценой контракта 5 на линии 1
- Потребитель 3 класса с ценой контракта 5 на линии 1
- Потребитель 2 класса с ценой контракта 3,0 на линиях 2 и 3
- Потребитель 1 класса с ценой контракта 5 на линиях 2 и 3
- Потребитель 1 класса с ценой контракта 5,3 на линиях 2 и 3
- Солнечная электростанция с ценой обслуживания 3 на линии 3
- Солнечная электростанция с ценой обслуживания 3 на линии 3
- 2 дизельные электростанции
- 1 аккумуляторная батарея

Основной управляющий скрипт:

```
def doTick():  
    # включаем все линии  
    for line in lines:  
        if (not lines[line]):  
            lineOn(line)
```

```

consumption = 0 # прогноз суммарного потребленияш

for H in houses:
consumption += H["forecast"][0]

for H in hospitals:
consumption += H["forecast"][0]

for H in factories:
consumption += H["forecast"][0]

generation = 0 # прогноз суммарной генерации

for G in wind_gens:
if G["online"]:
generation += G["value"]

for G in sun_gens:
if G["online"]:
generation += G["value"]

for C in contracts.active:
flux = C["flux"]
if (flux > 0):
generation += flux
else:
consumption -= flux

generation *= 0.85 #вычитаем 15%
consumption *= 1.15 # накидываем 15%
shortage = consumption - generation
shortage = consumption - generation
max_diesel = diesels["count"] * 5.99
charge_cells = cells["charge"]
capacity_cells = cells["count"] * 23.99
charge_speed_cells = cells["count"] * 3.59
discharge_speed_cells = cells["count"] * 7.19
if (shortage-8)>0:
proposeContract(1, 1, -0.7, 20)
if shortage<0:

```

```

        setCell( max ( -charge_speed_cells, max ( shortage, charge_cells -
capacity_cells )))
        shortage-=max ( -charge_speed_cells, max ( shortage, charge_cells -
capacity_cells ))
        if shortage>0:
            setCell(min(min(discharge_speed_cells, shortage), charge_cells) )
            shortage-=min(min(discharge_speed_cells, shortage), charge_cells)
            if (shortage<4.9) and (shortage>0.0):
                setExternal(shortage+0.01)
            shortage=0.0
        else:
            if shortage>0:
                setExternal(4.99)
            shortage-=4.99
            if (shortage < (max_diesel-0.01)) and (shortage>0.0):
                setDiesel(shortage+0.01)
            shortage=0.0 # ставим дизели на величину недостатка
        else:
            if shortage>0.0:
                setDiesel(max_diesel)
            shortage-=max_diesel
        else:
            setDiesel(0.0)
        for contract in contracts.pending:
            if contract["price"]<=-1:
                declineContract(contract)
            else:
                passContract(contract)

```

Запасной управляющий скрипт:

```

def doTick():
    # включаем все линии
    for line in lines:
        if (not lines[line]):
            lineOn(line)

    consumption = 0 # прогноз суммарного потребления

    for H in houses:

```

```

consumption += H["forecast"][0]

for H in hospitals:
consumption += H["forecast"][0]

for H in factories:
consumption += H["forecast"][0]

generation = 0 # прогноз суммарной генерации

for G in wind_gens:
if G["online"]:
generation += G["value"]

for G in sun_gens:
if G["online"]:
generation += G["value"]

for C in contracts.active:
flux = C["flux"]
if (flux > 0):
generation += flux
else:
consumption -= flux

generation *= 0.75 # вычитаем 25%
consumption *= 1.25 # накидываем 25%

shortage = consumption - generation
max_diesel = diesels["count"] * 5.99
charge_cells = cells["charge"]
capacity_cells = cells["count"] * 23.99
charge_speed_cells = cells["count"] * 3.59
discharge_speed_cells = cells["count"] * 7.19

if (shortage <= 0): # если избыток есть, он идёт в аккумулятор
setCell( max ( -charge_speed_cells, max ( shortage, charge_cells -
capacity_cells )))
setDiesel(0)
setExternal(0)

```

```
else:
    setCell(min(min(discharge_speed_cells, charge_cells), shortage))
    shortage-=min(min(discharge_speed_cells, charge_cells), shortage)
    if (shortage < max_diesel):
        setDiesel(shortage) # ставим дизели на величину недостатка
    else:
        setDiesel(max_diesel) # ставим дизели на полную
        setExternal(shortage - max_diesel) # остальное берём снаружи

# все контракты отправляются игроку
for contract in contracts.pending:
    passContract(contract)
```