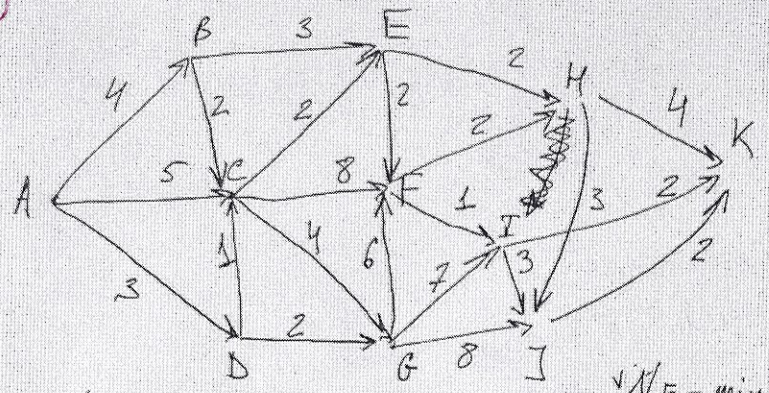
37б

Вариант №1

№4 Построим по таблице граф.

5б




$N_x$ - минимальный путь в город X

$N_A = 0$

$N_B = N_A + 4 = 4$

$N_D = N_A + 3 = 3$

$N_C = \min(N_A+5; N_B+2; N_D+1) = \min(5; 6; 4) = 4$

$N_E = \min(N_B+3; N_C+2) = \min(7; 6) = 6$

$N_G = \min(N_C+4; N_D+2) = \min(8; 3) = 5$

$N_F = \min(N_E+2; N_C+8; N_G+6) = \min(8; 12; 11) = 8$

$N_H = \min(N_E+2; N_F+2) = \min(8; 10) = 8$

$N_I = \min(N_F+1; N_G+7) = \min(9; 13) = 9$

$N_J = \min(N_H+3; N_I+3; N_G+8) = \min(11; 12; 13) = 11$

$N_K = \min(N_H+4; N_I+2; N_J+2) = \min(12; 11; 13) = 11$

Ответ: Из A → K придётся проехать 11 усл. ед.

Маршрут: $A \xrightarrow{3} D \xrightarrow{1} C \xrightarrow{2} E \xrightarrow{2} F \xrightarrow{1} I \xrightarrow{2} K$

№5 3б Пронумеруем канистры от 1 до 4.

Смешаем воду из ① и ②, тестируем её

химия обнаружена

Делаем тест ①

обнаруж. — ① - с химией

необнаруж. — ② - с химией

химия не обнаружена

Делаем тест ③

обнаруж. — ③ - с химией

не обнаруж. — ④ - с химией

№9 В ~~пр~~ задании не сказано, что программа должна завершаться, поэтому в программе использован бесконечный цикл. Программу можно завершить, нажав Ctrl+C. стр. 2

Вар 1

Задача на другой странице

Язык: Python 3.6

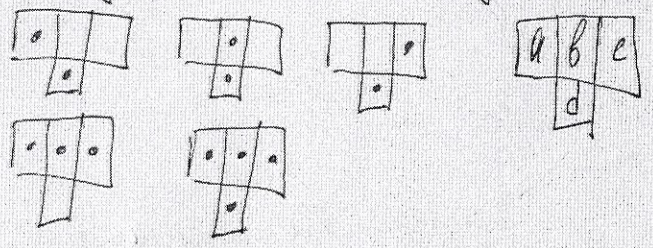
```
volumes = []
for i in range(sz):
    volumes.append({"hour": 0}
    volumes.append({"hour": 0, "minute":0, "data":0, "cycles": 0})
current=0
while True:
    cmd = input()
    if cmd == "add":
        hour = input("Введите время в часах")
        minute = input("Введите время в минутах")
        data = input(f"Время измерения {hour}:{minute}. Введите значение температуры")

        volumes[current]["hour"]=hour
        volumes[current]["minute"] = minute
        volumes[current]["data"] = data
```

5

№6 10

Все возможные комбинации:




| a | b | c | d | F |
|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

$$F=(a\wedge\bar b\wedge\bar c\wedge d)\vee(\bar a\wedge b\wedge\bar c\wedge d)\vee(\bar a\wedge\bar b\wedge c\wedge d)\vee(a\wedge b\wedge c\wedge\bar d)\vee$$

$$\vee(a\wedge b\wedge c\wedge d)=\cancel{d\wedge((a\wedge b\wedge\bar c)\vee(a\wedge b\wedge\bar c)\vee(\bar a\wedge\bar b\wedge c))\vee(a\wedge b\wedge c)}$$

$$=(a\wedge\bar b\wedge\bar c\wedge d)\vee(\bar a\wedge b\wedge\bar c\wedge d)\vee(\bar a\wedge\bar b\wedge c\wedge d)\vee(a\wedge b\wedge c).$$

Вариант 1



17 Формируется массив: 1, ..., 20

Внеш. цикл: пока посл. эл. > 10

10 Внутренний цикл „перегоняет" последний элемент, пока соседний элемент слева ≥ M.

результат

M=10 k=0 => 1, ..., 10, 20, 11, ..., 19 k+9

M=9,5 k=9 => 1, ..., 9, 19, 10, 20, 11, ... 18 k+10

M=9 k=19 => 1 ... 9, 18, 19, 10, 20, 11, ... 17 k+10

M=8,5 k=29 => 1, ..., 8, 17, 9, 18, 19, 10, 20, 11 ... 16 k+10

M=8 k=40 => 1, ..., 8, 16, 17, 9, 18, 19, 10, 20, 11, ... 15 k+11

~~аналогично для „15" и „14" => k+~~

M=7,5 k=51 => 1, ..., 7, 15, 8, 16, 17, 9, 18, 19, 10, 20, 11, 12, 13, 14 k+12

M=7 k=63 => 1, ..., 7, 14, 15, 8, 16, 17, 9, 18, 19, 10, 20, 11, 12, 13 k+12

M=6,5 k=75 => 1, ..., 6, 13, 7, 14, 15, 8, 16, 17, 9, 18, 19, 10, 20, 11, 12 k+13

M=6 k=88 => 1, ..., 6, 12, 13, 7, 14, 15, 8, 16, 17, 9, 18, 19, 10, 20, 11 k+13

M=5,5 k=101 => 1, ..., 5, 11, 6, 12, 13, 7, 14, 15, 8, 16, 17, 9, 18, 19, 10, 20 k+14

M=10 k=115

M=10 k=0 => 1, ..., 9, 20, 10, ..., 19 k+10

M=9,5 k=10 => 1, ..., 9, 19, 20, 10, ..., 18 k+10

M=9 k=20 => 1, ..., 8, 18, 9, 19, 20, 10, ..., 17 k+11

Далее аналогично

„17" => k+11 „14" => k+13 „11" => k+14

„16" => k+12 „13" => k+13 затем в конце 10

„15" => k+12 „12" => k+14 и программа завершается

$$\begin{array}{r} 24 \\ \times\ 5 \\ \hline 120 \end{array}$$

$$k = 2(10+11+12+13+14) = 2 \cdot \frac{10+14}{2} \cdot 5 = 120$$

Ответ: 120.

№ Python 3.6 180 Вар. 1 стр. 4

```
volumes=[], cycles=[]
for i in range(32):
    volumes[i] ~~=0~~, cycles[i]=0, 0
current=0
while True:
    cmd=input()
    if cmd == "add":
        hour=int(input("Введите время в часах"))
        minute= int(input("Введите время в минутах"))
        data=int(input(f"Время ~~в~~ измерения {hour}:{minute}.
                Введите значение температуры"))
        volumes[current]=hour*60+minute
        volumes[current+1]=data+50   cycles[current]+=1
        current=(current+2)%32       cycles[current+1]+=1
        print(f"Добавлено измерение: в {hour}:{minute} T={data}")
    elif cmd=="all":
        for i in range(32):
            if i%2==1 or volumes[i]==0:
                continue
            ~~print(f"~~
            hour=volumes[i]//60
            minute=volumes[i]%60
            print(f"{hour}:{minute} T={volumes[i]-50}")
    elif cmd=="~~test~~ cycles":
        cell=int(input("Введите номер ячейки"))
        print(f"Ячейка {cell} была перезаписана {cycles[cell]} раз")
    elif cmd=="test":
        count=sum(cycles)//2
        ~~print("В память можно записать~~
        left=str(16000-count)
        print(f"В память можно записать {left} измерений")
```

23ИМ-217

Вариант 1



№10 Python 3.6

```
~~field=[[~~ import random
field=[]
for i in range(8):
    field.append([0]*8)
distance=1
while True:
    cmd=input()
    if cmd=="show":
        for i in range(8):
            print(" ".join([str(x) for x in field[i]]))
    if cmd=="add":
        for i in range(7):        choose=[]
            for j in range(7):
                if field[i][j]==0 and field[i][j+1]==0 and
                field[i+1][j+1]==0 and field[i+1][j]==0:
                    choose.append((i,j))
        if not choose:
            print("Нет места для новой фигуры")
        else:
            x,y=random.choice(choose)
            field[x][y],field[x+1][y],field[x][y+1],field[x+1][y+1]=8,8,8,8
    if cmd=="distance":
        distance=int("Введите перемещение")/100
    if cmd="left":
        for j in range(1,8):
            for i in range(8):
                field[i][j-1]+=distance*field[i][j]
                field[i][j]-=distance*field[i][j]
```

↳ на след. странице

23ИН-217

№10 продолж. | Вар. 41 | №5 стр.

```
if cmd == "right":
    for j in range(7,-1,-1):
        for i in range(8):
            field[i][j+1] += distance * field[i][j]
            field[i][j] -= distance * field[i][j]
if cmd == "up":
    for i in range(1,8):
        for j in range(8):
            field[i-1][j] += distance * field[i][j]
            field[i][j] -= distance * field[i][j]
if cmd == "down":
    for i in range(7,-1,-1):
        for j in range(8):
            field[i+1][j] += distance * field[i][j]
            field[i][j] -= distance * field[i][j]
```