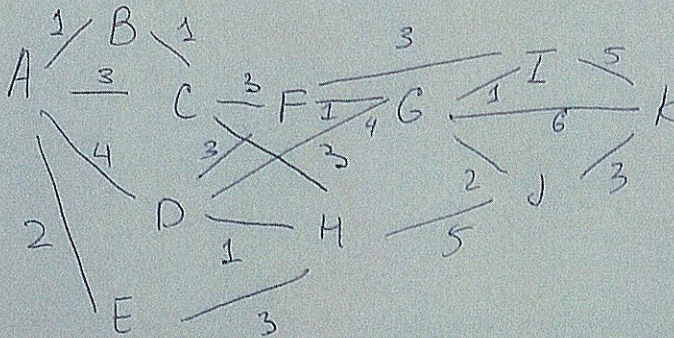
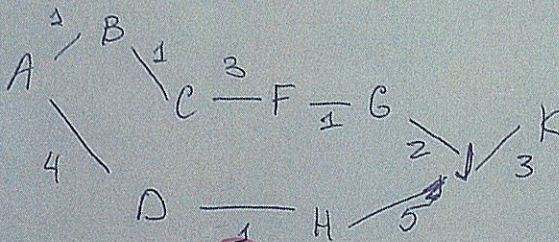


Задание 4 (6)

На основе матрицы составили граф

Компьютер 1, вар. 4.,
стр. 1.

Упростим его, удалив ребра, у которых есть более короткий альтернативный путь

Осталось два пути, из них самый короткий - ABCFGJK, длина $1+1+3+1+2+3=11$.

Задание 5 (7)

Воспользуемся графическим методом ёмкостями. В системе в 4 пустых ёмкостях помещаем воду из 4 канистр. (4 канистры - в одну, другие 4 - во вторую и т.д.). Далее проверим оставшиеся четыре ёмкости те 4 канистры, тест у нас ёмкости дал положительный результат.

Задание 3 (продвижение)

По 2 и 4 кв. человек возраста [10; 12] 41.

Задание 6 (5)

Пусть A - красный, B - белый, C - желтый, D - желтый розоватый, а F - функция светотерапии. Тогда запрещённые комбинации обозначены в следующей таблице истинности нулями.

A	B	C	D	F
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	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	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

В форме не декартовой логической функции это:

(A и НЕ B и

 $(A \wedge \bar{B} \wedge \bar{C} \wedge \bar{D}) \vee (\bar{A} \wedge B \wedge \bar{C} \wedge \bar{D}) \vee (\bar{A} \wedge B \wedge C \wedge \bar{D}) \vee (\bar{A} \wedge B \wedge \bar{C} \wedge D) \vee (A \wedge B \wedge \bar{C} \wedge \bar{D}) \vee$ $(A \wedge B \wedge C \wedge D) \vee (A \wedge \bar{B} \wedge \bar{C} \wedge D)$

Если упростить, то получим следующее:

 $(B \wedge \bar{C} \wedge \bar{D}) \vee (\bar{A} \wedge \bar{B} \wedge \bar{C}) \vee (D \wedge \bar{B} \wedge \bar{C}) \vee (A \wedge \bar{B} \wedge \bar{C} \wedge \bar{D})$

Т.е. нам нужны только запрещённые комбинации, применены "НЕ"

 $(B \wedge \bar{C} \wedge \bar{D}) \vee (\bar{A} \wedge \bar{B} \wedge \bar{C}) \vee (D \wedge \bar{B} \wedge \bar{C}) \vee (A \wedge \bar{B} \wedge \bar{C} \wedge \bar{D})$

Задача 9

18

Компьютер 1, Вар. 4.,
ср. 2.

```

class Meteo:
    hours = []
    minutes = []
    temperatures = []
    rec = []

    def __init__(self):
        for i in range(256):
            self.hours.append(0)
            self.minutes.append(0)
            self.temperatures.append(0)
            self.rec.append(0)

    def add(self):
        print("Введите время измерения в часах")
        h = int(input())
        print("Введите время измерения в минутах")
        m = int(input())
        print("Введите время измерения {h}; {m}. Введите значение температуры в градусах Цельсия")
        t = int(input())
        self.temperatures[self.i] = t
        self.hours[self.i] = h
        self.minutes[self.i] = m
        self.rec[self.i] += 1
        self.i += 1
        self.i %= 256
        if self.i == 0 and self.rec[-1] == 125:
            print("Временная память переполнена")

    def cycles(self):
        print("Введите номер цикла")
        x = int(input())
        print("Выводка {x} блока переписана {self.rec[x]} раз")

    def all(self):
        for j in range(256):
            if self.rec[j] != 0:
                print("{self.hours[j]}: {self.minutes[j]} T = {self.temperatures[j]}")

    def test(self):
        s = 256 * 125
        for j in range(256):
            s -= self.rec[j]
        print("В переполненную память временно записано {s} измерений", end=" ")
        if s % 10 == 1:
            print("измерение")
        elif s % 10 == 2 or s % 10 == 3 or s % 10 == 4:
            print("измерения")
        else:
            print("измерений")

meteo = Meteo()
while 1:
    action = input()
    if action == "add":
        meteo.add()
    elif action == "cycles":
        meteo.cycles()
    elif action == "all":
        meteo.all()
    elif action == "test":
        meteo.test()

```