

ИНФОРМАТИКА

предмет

ШИФР 6/18 ИИФ/12

Задание 1

Построим таблицу и заполним её:

корзина	ваза	банка	контейнер	
X	X	X	✓	сахар
X	X	✓	X	фрукты
✓	X	X	X	конфеты
X	✓	X	X	шоя

По условию известно, что сахар и шоя не в корзине, конфеты и сахар не в банке. Из условия ёмкости с конфетками между вазой и ёмкостью с фруктами => в вазе не конфеты и не фрукты. Также контейнер между банкой и ёмкостью с фруктами => фрукты не в банке и не в контейнере => в банке фрукты => шоя в корзине конфеты => в контейнере сахар => в вазе шоя. Это возможно, если ёмкости расположены по кругу:

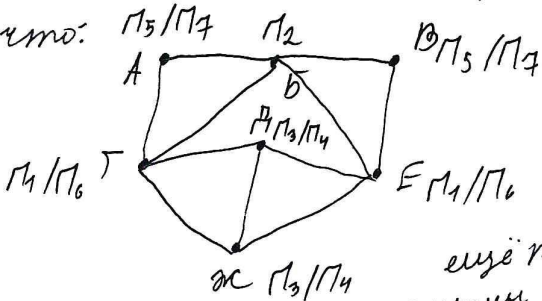


10 баллов

Ответ: в корзине конфеты, в банке фрукты, в контейнере сахар, в вазе шоя.

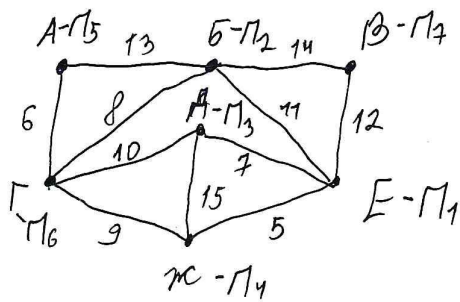
Задание 3.

Опираясь на таблицу и много ребёр у каждой вершины можно сказать, что:



по таблице однозначно можно определить точку вершину B, но ещё известно, что от A до M не более 15 км. От A до B кратчайший путь 13 или 14 км (вершина A ещё точно неизвестна), но от B до любой другой вершины не менее 4 км => кратчайший путь лежит не через вершину B => кратчайший путь проходит через G. Возможно два варианта для A и G: P5 и P6 или P7 и P1, расстояние между P5 и P1 уже 12 км, а от P1 до M (которая P3 или P4) не менее 5 км => точка G это P6, а точка A - P5. Тогда точкой M является либо P4 (так как от P6 до P4 меньше расстояние, чем до P3), тогда от A до M по кратчайшему пути равно 15 км.

Теперь можно однозначно связать графы таблицы:



Нам нужно определить кратчайшие расстояния от точки А: до Е 8 км (А-Е), до М 13 км (А-Е-М), до Г 10 км (А-Г), до В 5 км (А-Е-В или А-Г-В), до В тогда 19 км (А-Е-В). *р/р бабилев*

Ответ: кратчайший путь из А в В равен 19 км.

### Задача 5.

На языке программирования Python такую сортировку можно реализовать через стандартную сортировку и lambda-функцию, которую берет с двумя значениями: сумма цифр и само число, таким образом список отсортируется по сумме цифр каждого числа, а если суммы равны, то по самим числам. Также в коде дана функция 4 числа, такую сортировку можно написать и самому в более общем виде:

```
начало
сумма цифр числа a, b, c, d в отдельной переменной
если
функция СУММ_Ц (num):
    ans = 0
    x = str(num); // x - строка числа num
    цикл for i = 0 до длины(x):
        ans += int(x[i]) // преобразуем i-ую цифру числа (ans) обратно в integer.
    вернуть ans // функция возвращает ans
```

```
начало
символы a, b, c, d в отдельных переменных;
a_s = СУММ_Ц(a); b_s = СУММ_Ц(b); c_s = СУММ_Ц(c); d_s = СУММ_Ц(d);
если a_s > b_s то:
    (a_s, b_s) = (b_s, a_s); // меняем местами значения
    (a, b) = (b, a);
если (a_s == b_s) и (a > b) то:
    (a_s, b_s) = (b_s, a_s);
    (a, b) = (b, a);
если (c_s > d_s) или ((c_s == d_s) и (c > d)) то:
    (c_s, d_s) = (d_s, c_s);
    (c, d) = (d, c);
если (b_s > d_s) или ((b_s == d_s) и (b > d)) то:
    (b_s, d_s) = (d_s, b_s);
    (b, d) = (d, b);
если (a_s > c_s) или ((a_s == c_s) и (a > c)) то:
    (a_s, c_s) = (c_s, a_s);
    (a, c) = (c, a);
```

ИНФОРМАТИКА

предмет

ШИФР 6118ИИФ12

предложение  
задания 5

вывод a, c, b, d;  
команды.

~~команды~~

Задание 6.

решение на языке программирования python 3.10:

```
goods = {"A": 0, "B": 0, "C": 0, "D": 0, "E": 0, "F": 0}
N = int(input())
for _ in range(N):
    code, x = input().split()
    goods[code] += int(x)
print(*goods.values())
```

временная сложность  $O(N)$ . Запись весов товаров реализуется через словарь, так как товары имеют буквенные индексы.

Задание 7.

решение на языке программирования Python 3.10:

```
from math import factorial
N, M, K = map(int, input().split())
ans = pow(N, M)
for x in range(0, K):
    ans -= pow(N-1, M-x) * factorial(M) // (factorial(x) * factorial(M-x))
print(ans)
```

временная сложность  ~~$O(N^M)$~~  алгоритма  $O(K)$ . Вместо перебора всех слов, пересчитав можно ускорить при помощи комбинаторных формул. Все слов возможно  $N^M$ , но среди них есть такие, где число букв A меньше K. Вычтем такие слова из результата. Пусть x - число A, где  $x \leq K$ , тогда слов с x буквами A будет  $\binom{N-1}{M-x} \cdot \frac{M!}{x!(M-x)!}$ , где  $\binom{N-1}{M-x}$  - число способов выбрать  $M-x$  букв из всех, кроме A.  $\frac{M!}{x!(M-x)!}$  - число сочетаний,

то есть количество способов выбрать x букв ~~из~~ ~~среди~~ ~~этих~~ букв в слове из N букв, порядок не учитывается (1, 3, 4 будет тоже, что и 3, 4, 1).