

12 M

Московский государственный технический университет имени Н.Э. Баумана

Шифр 112971
(заполняется ответственным секретарем приемной комиссии)

ПИСЬМЕННАЯ РАБОТА
на олимпиаде «Шаг в будущее»

соревнования по образовательному предмету информатика
(наименование дисциплины)

«Шаг в будущее»

Фамилия И. О. участника СВИРИНА ДАРЬЯ ДМИТРИЕВНА

Город, № школы (образовательного учреждения) г. Москва, лицей 1525

Регистрационный номер ШМ0038

Вариант задания для 10 класса

Дата проведения « 12 » марта 2016 г.

Подпись участника В.В.

1	2	3	4	5	6	7	8	9	10	Σ
12	12	12	12	22						70

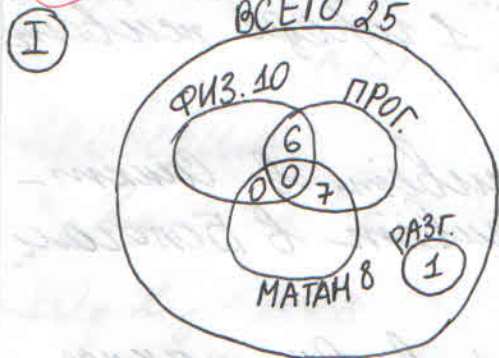
Шифр 112971

(заполняется ответственным секретарем приемной комиссии)

2971

Вариант № Для 10 класса

УЗ.



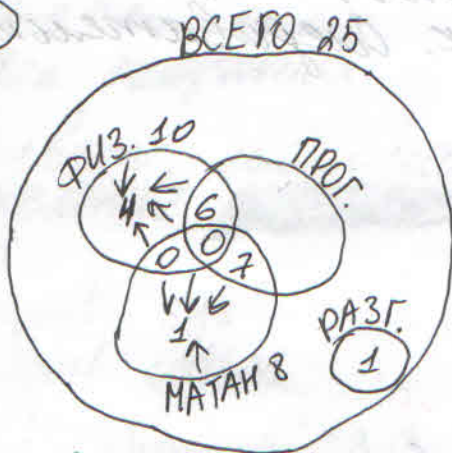
1) Тогда тех, кто сделал домашнее задание только по программированию останется:

$$25 - 1 - 4 - 6 - 7 - 1 = 6$$

2) Тогда тех, кто сделал домашнее задание только по одному предмету останется:

$$4 + 6 + 1 = 11 + 128$$

II



Ответ: 11 человек выполнили задание строго по одному предмету.

УН.

□□₈ ← Максимум каждого разряда = 7
Величи на 2 и получаем 3,5, округляем до 3.

Таким образом, максимальное значение суммы двух соседних разрядов трехзначного числа в шестнадцатеричной системе счисления = 3

В таких условиях максимальное число = 303₁₆

Ответ: максимальное число = 303₁₆

В2.

1) + 2) Пассажиры живут в Москве, Санкт-Петербурге и Беломом, т.е. в городе живет только 1 пассажир.

4) пассажир Иванов - Москва

3) + 5) живут в Беломом.

7) пассажир Петров живет не в Беломом, т.к. там живет вся бригада, включая его одногруппника.

пассажир Петров не может жить в Москве, т.к. там живет пассажир Иванов, а в 1 городе живет 1 пассажир.

Следовательно, пассажир Петров живет в Санкт-Петербурге, а оставшийся Сидоров живет в Беломом.

6) проводник - Петров

8) помощник машиниста - не Сидоров. Он так же не Петров, т.к. Петров - проводник. Следовательно, помощник машиниста - Иванов.

Оставшийся машинист - Сидоров.

Ответ:

Бригада:

- все живут в Беломом

- машинист - Сидоров

- помощник машиниста - Иванов

- проводник - Петров

Пассажиры:

Сидоров - Беломое

Иванов - Москва

Петров - Санкт-Петербург

125

11.

$$3yxy + 5yxy = 20y \quad x - \text{целое число}$$

В десятичной это:

$$3x_{10} + 5x_{10} = 2y \rightarrow 8x_{10} = 2y$$

x -целое. При 0 число не будет, т.к. не составим сс.

Возьмем $x_{10} = 1$, тогда $y = 4$; — этого не может быть, т.к. в уравнении присутствует цифра 5, а её не может быть в четверичной сс.

Возьмем $x_{10} = 2$, тогда $y = 8$

$$x_{10} = 2 \Rightarrow x_8 = 2$$

Проверим:

$$3_8 x_8 + 5_8 x_8 = 20_8$$

$$10_8 x_8 = 20_8$$

$$x_8 = 2$$

Всё сходится.

⊕ 125.

Ответ: минимальное основание системы счисления ~~равно 8~~ равно 8.

15.

```
import sys
import codecs

def shortestpath(graph, start, end, visited = [], distances = {}, predecessors = {}):
    # если пришли в конечн. вершн, находим весь путь
    # и возвращаем результат
    if start == end:
        path = []
        while end != None:
            path.append(end)
            end = predecessors.get(end, None)
        return path[::-1]
    # если ни одна вершина не была пройдена, выставим
    # текущую длину пути = 0
```



```

if not visited: distances[start] = 0
# если считать графы один раз и не изменять
# если не посещен, то считать расстояние
for neighbor in graph[start]:
    if neighbor not in visited:
        neighbor_dist = distances.get(neighbor, 999999)
        tentative_dist = distances[start] + graph[start][neighbor]
        if tentative_dist < neighbor_dist:
            distances[neighbor] = tentative_dist
            predecessors[neighbor] = start
# если посещен, то считать расстояние
# если не посещен, то считать расстояние
visited.append(start)
# найти ближайшую к mapmy точку
unvisiteds = dict((k, distances.get(k, 999999)) for
k in graph if k not in visited)
closest_node = min(unvisiteds, key=unvisiteds.get)
# если мы нашли ближайшую точку и перешли к ней
return shortest_path(graph, closest_node, end, visited,
distances, predecessors)
# считаем файл с координатами
f = codecs.open("input.txt", "r", "utf-8")
data = f.read().split('\r\n')
f.close()
# считаем координаты
crosses_num = int(data[0])
start = data[1]
finish = data[2]
graph = {}
for read in data[4:]:
    read_data = read.split()
    graph[read_data[0]] = int(read_data[1])
    graph[read_data[1]] = int(read_data[2])

```


1	2	3	4	5	6	7	8	9	10	

Шифр 112971

(заполняется ответственным секретарем приёмной комиссии)

Вариант № для 10 класса

продолжение номера 5

```
print(shortest_path(graph, start, finish))
```

Я выбрала 999999, как дефолтное значение длины пути от точки до точки, если между ними нет дорог. Это нужно, чтобы легко находить ~~ближайшую~~ ближайшую точку. Можно было поставить и -1 и делать доп. проверку, но мне это показалось лишним и зря перепутался код.