**algorithmization and programming**

**final exam var 2**

ACTIVITIES

Напишите программы согласно вашему варианту

**Сохраните все коды задач в одном txt файле. Назовите файл вашим именем, фамилией и вариантом и загрузите в платонус (ex. Balgabek\_Askar\_2variant.txt)**

**Обратите внимание, после загрузки файла нужно дважды подтвердить его отправку. Необходимо отправить на проверку, иначе ваш вариант не будет виден проверяющему преподавателю**

Grades: One task – 50 points, two tasks – 80 points, three tasks – 100 points

Также, вы можете решать задачи в стиле ооп или через структуры. В случае решения через структуры, вы теряете 5 баллов с каждой задачи. Решение в стиле ооп подразумевает что все действия для решения задач необходимо поместить во внутренние функции классов. Также, следуйте принципам инкапсуляции (геттеры и сеттеры).

**Variant1**

1. Создайте структуру/класс **Dungeon**{

**name** – название подземелья

**type** – тип подземелья

**cost** - стоимость исследования

**income** – количество магических камней которые вы добудете после исследования

}

Ваша компания занимается разведкой древних подземелий с магическими камнями

Введите переменную **budget** – количество ваших денег для разведки подземелий. Введите количество подземелий, затем введите массив из подземелий.

Вы по очереди исследуете подземелья из массива. Каждое подземелье имеет тип (**type** - “Ancient”, “Mistery”, “Evil”). Также, каждое подземелье имеет стоимость (**cost**) - сколько нужно денег для его исследования, и доход (**income**) – сколько прибыли принесет его исследование.

Если тип подземелья “**Ancient**” вы исследуете его в любом случае.

Если тип подземелья "**Mistery**" то вы исследуете его только в случае если прибыль проекта (**income**) больше чем его стоимость (**cost**).

Если тип проекта "**Evil**" то вы не исследуете его

На исследование подземелья вы тратите свой бюджет (**budget**). Каждое подземелье приносит некоторое количество магических камней (**income**).

После того как закончится бюджет или закончатся подземелья – выведите общую сумму магических камней, и название всех подземелий, которые вы исследовали.

(совет1: используйте вектора. В них удобно добавлять принятых кандидатов)

**Example:**

**Input:**

10000

7

Darkhold Ancient 2000 1000

Desermore Evil 1000 100000

Stonehendege Mistery 4000 5000

Arivata Mistery 3000 2000

Ladaraka Ancient 3000 3000

Dadarada Ancient 2000 5000

Kagu Mistery 2000 20000

**Notes:**

Darkhold стоит 2000 и принес 1000 камней. В бюджете осталось 8000. Desermore имеет тип Evil и мы его пропускаем. Stonehendege имеет тип Mistery, но мы получим 5000 камней при расходе в 4000, значит его исследуем. В бюджете осталось 4000, а в банке 6000 камней. Arivata имеет тип Mistery и приносит меньше чем стоит, его тоже пропускаем. Ladaraka подходит, но после него в бюджете остается 1000 камней и на следующее подземелье уже не хватает.

**Output:**

9000

Darkhold

Stonehendege

Ladaraka

1. Создайте структуру/класс **Department**{

**name** – название департамента

**open\_time** – время открытия департамента

**max\_visitors** - максимальное количество посетителей за день

}

Создайте структуру/класс **Visitor**{

**name** – имя посетителя

**visit\_time** – время посещения в формате “hh:mm”

**destination** - название департамента куда он направляется

}

Вы вахтер в министерстве мистических искусств. Введите количество отделов министерства и создайте массив из министерств (**Department**). После введите количество посетителей и создайте массив из посетителей (**Visitor**). Это два отдельных вектора/массива.

У вас есть список отделов министерства. Каждый отдел имеет название (**name**), время открытия (**open\_time**) и максимальное количество посетителей (**max\_visitors**), которое оно может принять за день. Также у вас есть список посетителей. Каждый посетитель имеет имя (**name**), время прибытия (**visit\_time**) и название отдела куда он направляется (**destination**).

Вам нужно распределить посетителей по отделам, исходя из названия отдела, куда отправляется каждый посетитель (**destination**). Если посетитель пришел (**visit\_time**) раньше, чем открывается его отдел **(open\_time)** его направляют в комнату ожидания. Также если в отделе уже максимальное количество посетителей (**max\_visitors**), всех остальных кто направляется в него также отправляют в комнату ожидания.

Выведите имена и направления всех, кто попал в комнату ожидания.

(ВАЖНО1: Если делайте структуры, то теряете 5 баллов)

(ВАЖНО: сделайте в стиле ООП чтобы получить максимум баллов. Для этого используйте классы, и все вычисления должны быть в функциях внутри классов, также следуйте принципу инкапсуляции)

(ВАЖНО: в выводе порядок имен НЕ ВАЖЕН)

(совет1: используйте вектора. В них удобно добавлять принятых кандидатов)

**Example:**

**Input:**

2

Solar 14:00 3

Moon 10:30 2

8

Smith 13:00 Moon

Nurbek 9:00 Solar

Samara 10:15 Moon

Kaleb 20:00 Moon

Hiroshi 19:00 Solar

Deizi 11:00 Moon

Ralph 8:30 Solar

Mellisa 15:15 Solar

**Notes:**

Smith идет в Moon в 13:00, там прием с 10:30, значит он попадает туда

Samara идет в Moon в 10:15, там прием с 10:30, значит она идет в зал ожидания

Kaleb попадает в Moon

Deizi тоже идет туда вовремя, однако в Moon уже было 2 посетителя, а значит она идет в зал ожидания

**Output:**

Nurbek Solar

Samara Moon

Deizi Moon

Ralph Solar

1. Создайте структуру/класс **Order**{

**name** – название ордена

**magic\_type** – вид магии, используемый орденом

**necessary\_power** – необходимое количество энергии чтобы орден не уничтожился

**wizard\_number** – количество магов

**wizards** - массив/вектор из магов

}

Создайте структуру/класс **Wizard**{

**name** – имя мага

**experience** – количество лет опыта которые есть у мага

**specialization** – вид магии, на которой специализируется маг

**amulet\_power** – сила амулета который создает маг

}

Существуют N орденов магов, которые создают волшебные амулеты для предотвращения гибели планеты.

В каждом ордене есть своя специализация амулетов (**magic\_type**), которые вырабатывают энергию, и маги, которые создают эти амулеты (**wizards**). Также каждый орден потребляет определенной количество энергии для своего существования (**necessary\_power**).

Каждый маг создает амулеты определенной силы (**amulet\_power**). Если специализация мага (**specialization**) не совпадает со специализацией ордена (**magic\_type**), сила его амулетов в 2 раза слабее. Если опыт мага (**experience**) больше 10ти лет, то сила его амулетов в 2 раза сильнее. Если его опыт больше 30 лет, то его амулеты не слабеют даже если специализация не совпадает, а также в 2 раза сильнее поскольку его опыт больше 10 лет.

Силы амулетов каждого мага в ордене суммируются, и тратятся на поддержание существования ордена. Каждому ордену нужно определенное количество сил чтобы не погибнуть (**necessary\_power**). Если общего количества магии больше чем нужно ордену, его излишки идут в общий резервуар магии, который тратится на поддержку других орденов, у которых магии недостаточно.

Если общей магии не хватит на поддержание всех орденов, то планета погибнет. Ответьте на вопрос достаточно ли сил производят волшебники чтобы все ордена существовали, и если да, то сколько сил осталось в общем резервуаре.

(ВАЖНО1: Если делайте структуры, то теряете 5 баллов)

(ВАЖНО: сделайте в стиле ООП чтобы получить максимум баллов. Для этого используйте классы, и все вычисления должны быть в функциях внутри классов, также следуйте принципу инкапсуляции)

**Example:**

**Input:**

3

SunOrder Fire 600 4

Rundolf 5 Fire 100

Hembledor 9 Water 150

Kediv 51 Earth 200

Verlin 19 Water 100

SeaOrder Water 300 5

Quatua 15 Water

Lakaso 20 Fire

Unoloko 5 Earth

Akama 8 Water

Ri 31 Earth

MountainOrder Earth 400 3

Demi 25 Earth

Hemi 3 Earth

Temi 33 Earth

**Notes:**

SunOrder специализация Fire. Сила амулета Rundolf 100, и его специализация тоже Fire. Также его опыт 5 лет, значит без бонусов. По итогу Rundolf +100 к силе. Hembledor обладает силой 150, однако его специализация Water и стаж меньше 10ти лет, что значит что итоговая сила +75. Kediv с силой 200, его опыт 51 год, что значит что мы не берем в расчет что его специализация Earth. Также сила его амулета в 2 раза выше поскольку его опыт больше 10ти лет. Итог + 400. Verlin обладает Water, что значит что сила его амулета в 2 разаь меньше. Однако он маг больше 10ти лет, что значит что сила в 2 раза больше. По итогу, сила его амулета не меняется и составляет + 100. В общем, SunOrder производит 675 силы, а нужно 600. Значит орден цел и добавляет в общий котел +75 силы(75). SeaOrder произвел 305 силы из 300. +5 в общий котел(80). MountainOrder произвел 370 силы, однако нужно 400. Недостаток в -30 силы. Однако в общем котле +80 силы, и этого хватит чтобы покрыть недостаток и в итоге в общем котле осталось +50 и все ордена сохранились.

**Output:**

50