Story:

Fardin is a student in Glenforest Secondary School but for the past 2 years, he hasn’t cared about Christmas time at Glen. But times have changed and Fardin now embraces the school spirit set up by SAC. However, Fardin is too lazy to do all the work of Christmas time, especially the planning. Help Fardin get ready for Christmas!

Look at Fardin’s decorations! https://www.youtube.com/watch?v=z2GXuhQu8Jo

J1: Festive Fardin

As you know, Christmas time is filled with many colours but the main colours are ‘red’, ‘green’, and ‘white’. As Fardin is walking around school, he looks at people’s shirt and pants colours and wants to determine whether they’re suitably dressed for Christmas. If their shirt and pants colours are part of the Christmas colours, output “Jingle Bells”. If they don’t, output “boring…”

**Input**

Line 1: Their shirt colour, all lowercase

Line 2: Their pants colour, all lowercase

**Output**

Either “Jingle Bells” or “boring…”

**Sample Input**

purple

red

**Sample Output**

boring…

**Sample Input**

green

green

**Sample Output**

Jingle Bells

shirt = raw\_input()

pants = raw\_input()

if shirt == 'green' or shirt == 'red' or shirt == 'white':

if pants == 'green' or pants == 'red' or pants == 'white':

print "Jingle Bells"

else:

print "Boring..."

else:

print "Boring..."

J2: Purchasing Presents

Fardin is planning to buy presents for all his teachers. However, he only has ***C***, dollars! Given ***N*** number of presents Fardin wants to buy, output how much money he will have left. If he runs out of money, output “Fardin’s broke“

**INPUT**

Line 1: N, number of presents

Line 2: C

N Lines of presents

**OUTPUT**

How much money Fardin has left

**Sample Input**

4

20.00

2.45

6.54

3.19

1.44

**Sample Output**

6.38

**Sample Input**

2

5.00

3.14

2.71

**Sample Output**

Fardin’s broke

# But in real life, Fardin’s always broke

presents = int(input())

money = float(input())

for i in range(presents):

cost = float(input())

money -= cost

money = round(money,2)

if money < 0:

print ("Fardin's broke")

else:

print (format(abs(money),'.2f'))

J3: Christmas Presents

Now that Fardin has bought his presents, he must distribute them amongst his teachers. However, he is doing well with some teachers and poorly with some others. Fardin has bought ***P*** presents, each with an unique price value ***PC*** (0<=PC <= 64000). Fardin has ***T*** (2 <= T <= P <= 20) teachers who he likes with an unique rating of ***TR***. (1<= ***TR*** <= 100). However, he is too lazy to determine the distribution so he decides to ask you to code a program for him that distributes the most expensive presents to the teachers he likes the most and the worst presents to the teachers he doesn’t like. You don’t have to account for plural objects (i.e. a books). Output the teacher assignments in the form *teacher* will get a *present.*

**INPUT**

Line 1: integer, P

Line 2: integer, T

Following 2P Lines:

Present name

Present Cost, ***PC***

Following 2T Lines:

Teacher Name

Teacher Rating, ***TR***

**OUTPUT**

The teacher and the present they get in descending order.

**SAMPLE INPUT**

4

4

Compact Disc

2.50

Science Textbook

41.50

Chocolate Bar

0.99

Blu-ray Player

25.98

Mr. Nikoletos

93

Ms. Brown

90

Mr. Fong

92

Mr. Wilson

95

**Sample Output**

Mr. Wilson will get a Science Textbook

Mr. Nikoletos will get a Blu-ray Player

Mr. Fong will get a Compact Disc

Ms. Brown will get a Chocolate Bar

**Explanation**

Fardin loves Mr.Wilson the most, thus he gets the most expensive present, the Science Textbook. Next is Mr. Nikoletos, so he’ll get the second most expensive present, the Blu-ray player and so on.

presents = int(raw\_input())

teachers = int(raw\_input())

ps = []

ts = []

for i in range(presents):

a = raw\_input()

b = float(raw\_input())

ps.append([b,a])

for i in range(teachers):

a = raw\_input()

b = float(raw\_input())

ts.append([b,a])

ps.sort()

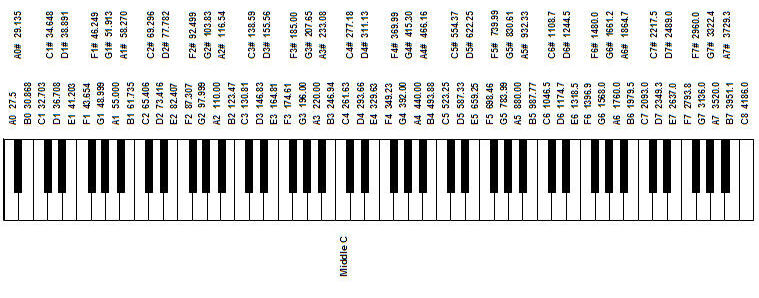
ts.sort()

for i in range(len(ts)):

print ts[i][1],'will get a',ps[i][1]

J4: Christmas Music

Fardin is looking for his first kiss and Christmas time would be the perfect time to get it! In preparation, he tries to compose his own tunes to play for the lovely Glenforest ladies who have an excellent taste in music. Unfortunately, Fardin has not learned simple harmony yet so he is essentially scribbling down notes. For his melody, he avoids using black keys for the sake of simplicity. The keys of the piano are labelled as below.



To judge whether Fardin’s melody is good or not, we have some rudimentary rules:

1. After a leap of a fifth or more (i.e. from C4 to G4), the following note cannot continue in the same direction (i.e. cannot be A5 or higher).
2. After a leap of a fifth or more, the following note cannot be another leap of a fifth or more.
3. There cannot a leap of more than an octave (i.e. C4 to D5)

Given a melody consisting of ***N*** (2<=***N***<=100) notes, output “Melodious!” if it is good or “Salieri’s music” if it is bad.

**Input**

Line 1: N, the number of notes

N Lines: a note in the form above

**Output**

“Melodious!” or “Salieri’s music”

**Sample Input**

10

E4

F4

C4

D4

D4

F4

G4

B4

D5

C5

**Sample Output**

Melodious!

**Sample Input**

5

G4

C4

F3

A3

C4

**Sample Output**

Salieri’s music

**Explanation 1:**

All the distances are less than a fifth so the melody is fine

**Explanation 2:**

The leap of five from G4 to C4 is followed another leap in the same direction, from C4 to F3, violating both rules 1 and 2.

trans = {'C':0,'D':1,'E':2,'F':3,'G':4,'A':5,'B':6}

notes = []

for i in range(int(raw\_input())):

note = raw\_input()

value = int(note[1])\*7+trans[note[0]]

notes.append(value)

good = True

leap = False

up = None

prev = notes[0]

for T in range(1,len(notes)):

diff = notes[T]-prev

change = abs(diff)

if diff > 0:

direction = True

elif diff == 0:

direction = None

elif diff < 0:

direction = False

if change > 7:

good = False

break

if change >= 4 and leap:

good = False

break

if leap and direction == up:

good = False

break

if change >= 4:

leap = True

else:

leap = False

up = direction

prev = notes[T]

if good:

print "Melodious!"

else:

print "Salieri's music"

**J5: Secret Santa**

Fardin is a secret Santa! Every Christmas, Fardin dresses up as a polar bear and spreads his good will to homes which do not chimneys. He accomplishes this by throwing presents through the windows (sometimes the windows are not open). This Christmas, Fardin comes across a particularly interesting building. Approached from the side, the building has only one column, and is 100 stories tall. There are windows at certain floors, and Fardin has bought specific gifts for each of these floors with windows. Every gift i is for floor F\_i and has a certain weight W\_i. Fardin starts on the roof, and with his superhuman strength, can move to one floor up or down every second (from the roof, it takes one second to move to floor 100). Fardin also requires one second to throw a present through a window. Furthermore, every second Fardin’s total stress increases by the sum of the gifts that he has not yet delivered, or is currently throwing through a window. Obviously, Fardin does not like to be stressed out while wearing a bear costume hanging from a building, so he asks you to find an order for him to deliver the presents so that he experiences the least stress.

**Input**

Line 1: one integer N, the number of presents Fardin must deliver (1 <= N <= 7)

Next N lines: for each line i, there contains two integers, F\_i and W\_i (1 <= F\_i <= 100)

Note: Fardin does not necessarily need to deliver presents in the order given in the input

**Output**

Output one integer, the minimum amount of stress Fardin must bear to deliver all the presents

**Sample Input**

3

100 1

1 200

2 1

**Sample Output**

20505

**Explanation for Sample Output**

There are three presents Fardin must deliver. One must go to floor 100 and has weight 1, one must go to floor 1 and weighs 200, and one must go to floor 2 and weights 1. In the optimal strategy, Fardin goes directly to floor 1, taking 100 seconds, and throws the present through in another second. The total stress delivering the first present is (1 + 1 + 200) \* (100 + 1). Next, Fardin moves to floor 2 and throws the present in. This adds (1 + 1) \* (1 + 1) stress. Finally, Fardin goes to floor 100 and delivers his last present for (1) \* (98 + 1) more stress. The total stress is 20505.