

Problem statement- Ali and helping innocent

- Arpasland has surrounded by attackers. A truck enters the city. The driver claims the load is food and medicine from Iranians. Ali is one of the soldiers in Arpasland. He doubts about the truck, maybe it's from the siege. He knows that a tag is valid if the sum of every two consecutive digits of it is even and its letter is not a vowel. Determine if the tag of the truck is valid or not.
- We consider the letters "A","E","I","O","U","Y" to be vowels for this problem.
- Input Format
 - The first line contains a string of length 9. The format is "DDXDDD-DD", where D stands for a digit (non zero) and X is an uppercase english letter.
- Output Format
 - Print "valid" (without quotes) if the tag is valid, print "invalid" otherwise (without quotes)
- SAMPLE INPUT
 - 12X345-67
- SAMPLE OUTPUT
 - invalid
- Explanation
 - The tag is invalid because the sum of first and second digit of it is odd (also the sum of 4'th and 5'th, 5'th and 6'th and 8'th and 9'th are odd).

In [1]:

```
1 # Ali and helping innocent people
2 n=input()
3 if len(str(n))==9:
4     if ((int(n[0])+int(n[1]))%2 == 0 and (n[2]!='A' and n[2]!='E' and n[2]!='I' and n[2]!='O' and n[2]!='U' and n[2]!='Y')):
5         print("valid")
6     else:
7         print("invalid")
```

15S464-73
valid

In [7]:

```

1  t = int(input())
2  for i in range(1,t+1):
3      m = int(input())
4      total = 0
5      for i in range(1,m+1):
6          ld=input().split()
7          l=int(ld[0])
8          d=int(ld[1])
9          sum = 0
10         for j in range(1,l+1):
11             sum = sum + d
12         total = total + sum
13     print(total)
14     if total>9:
15         sum1 = 0
16         def sune()
17         while(n>9):
18             r = total % 10
19             total = total / 10
20             sum1 = sum1 + r
21         if sum1<=9:
22             print(sum1)
23
24

```

```

1
3
1 2
2 1
2 9
22

```

Type *Markdown* and LaTeX: α^2

```
In [32]: 1 # Function to add contact to contacts text file
2 #from Packages import validators as v
3 from Packages.validators import phoneNumberValidator as pnv, emailValidation
4 #from Packages.validators import emailValidation as ev
5 def addContact(name,phone,email):
6     # store data as name,phone,email in the contacts file
7     filename = 'DataFiles/contacts.txt'
8     if not checkContactExists(name):
9         if pnv(phone) and ev(email):
10             with open(filename, 'a') as f:
11                 line = name + ',' + str(phone) + ',' + email + '\n'
12                 f.write(line)
13                 print(name, 'adeed to contacts')
14             else:
15                 print("Invalid Phone number and Email")
16                 return
17         else:
18             print(name, 'already exist')
19         return
20 #addContact("name1",98765543221,"name1_123@gmail.com")
21
22 import re
23 # Function to check if contact already exists
24 def checkContactExists(name):
25     filename = 'DataFiles/contacts.txt'
26     with open(filename, 'r') as f:
27         filedata = f.read() # f.read take all data as string
28         pattern = name+', '
29     return re.search(pattern, filedata)
30
31     #if checkContactExists(name):
32     #    return True
33     # else:
34     #    return False
35
36 #addContact('name4',123444,'addd')
37 addContact('name4',8976543210,'name4_123@gmail.com')
38
```

Invalid Phone number and Email

```
In [35]: 1 filename = 'DataFiles/contacts.txt'
2 def csvToList(filename):
3     li = []
4     with open(filename, 'r') as f:
5         for line in f:
6             li.append(line.split(','))
7     return li
8 #csvToList(filename)
9
10 def listToFile(li):
11     s = ''
12     for i in li:
13         s += ','.join(i)
14     return s
15 li = csvToList(filename)
16 listToFile(li)
```

```
Out[35]: 'name1,9876554320,name1_234@gmail.com\nname2,8765432109,name2_345@gmail.comname
1,98765543221,name1_123@gmail.com\n'
```

```
In [ ]: 1
```