15-110 Refresher Session : Week 14

No Calculators, only Brains !!

1. **Short Answers: One Line Code**

   (a) Complete the function `numOfWords(S)` that takes a given string `S` and returns the number of words in that string using one line. The only separator used in the string is space. For example `numOfWords("I am a cat")` returns `4`

   ```
   def numoOfWords(S) :
   ```

   (b) Complete the function `wordReverse(S)` that given a string `S` returns a string with words in reverse order. The only separator using in the string is space. For example, `wordReverse("I am a cat")` returns `"cataamI"`.

   ```
   def wordReverse(S):
   ```

   (c) Complete the function `sumSecondHalf(L)` that given a list `L` of numbers returns the sum of the second half of the list. If there are odd number of elements in the list, it should include the middle number. For example, `sumSecondHalf([1,2,3,4])` returns `7`

   ```
   def sumSecondHalf(L):
   ```

   (d) Complete the function `minKeys(d)` which given a dictionary `d` returns the largest key present in the dictionary. For example, `minKeys({ 3 : 2 , 2 : 4 , 3 : 6 , 4 : 8)` returns `2`.

   ```
   def minKeys(d):
   ```

2. **Act like a Computer** For the following functions, write what it prints in space provided

(a)
```
def mystery1(a , b ) :
        return (a + b) % 17  + (a - b) // 2  + a**2
a = 4
b = 5
print(mystery1( b , a))
```

(b)
```
def mystery2(L):
    k = []
    for b in L:
        k.append(b[-1])
    return dict(zip(k,k))

print(mystery2([[2] , [30 ,14]]))
```

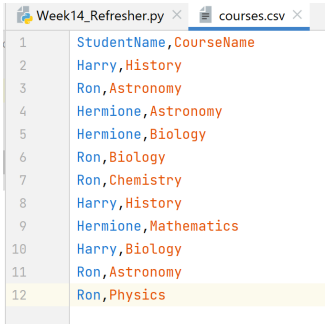
(c)
```
def mystery3():
    l = [0 , 1 , -1 , 2]
    d = {'2' : 2 , '0' : 0}
    s = set()

    for i in l :
        s.add(l[i] + len(s))
        s.discard(i+1)
    s.update(d)
    return s
print(mystery3())
```

3. **Act like a Programmer**

(a) Write the function `courses(filename)` that reads a file with students and courses as rows from a CSV file and prints the courses and the number of course that each student takes. The name of the CSV file is passes as input in the string `filename`. An example of sch a file is shown below where the first field is a student name and the second field is a course name. Fields are separated by commas, but extra spaces can be there.The function implemented on the file below should print

```
{'Harry': ['Biology', 'History'],
'Ron': ['Biology', 'Chemistry', 'Physics' ,'Astronomy'],
'Hermione': ['Biology', 'Mathematics', 'Astronomy']}
```



(b) Write a function `alternates(s)` which takes in a string S and puts the alternate letters to two different strings and returns both strings as a tuple. If both the strings are the same, your tuple should contain only one representation of it. For example `alternates('career')` returns `('cre', 'aer')` and `alternates('aaaa')` returns `('aa')`

Challenge 1 : What if you wanted to return only the unique characters in each string as a tuple ?
Challenge 2 : Let's say that the function is given as `alternates(s, n)` where n is the number

of alternates you need to have from the string for example - the word career with n = 3 gives `('ce', 'ae', 'rr')`

4. Questions that you can practice on include but are not limited to :
   (a) **Primes** - isPrime ? NthPrime ? Largest Prime ? Smallest Prime? Second Largest, etc.
   (b) **Greatest Common Divisor**
   (c) **Anagrams**
   (d) **Palindromes** - numeric, list, strings
   (e) **Substrings** - length ?