**BCSE101E Computer Programming: Python**

**Midterm Assessment – 2**

**Q1.**

You are given a sentence that contain some Voter ID card numbers. Write a program using **Regular expression** (MUST) to find and print all the valid Voter ID card numbers in the sentence.

A valid voter ID card number is a 10-length alphanumeric word that is:

* A combination of **alphabets (A to Z)** and **digits (from 1 to 8) only.**
* Voter ID allows only Uppercase letters.
* Voter ID starts with an alphabet and ends with a digit.
* Only the first four characters are alphabets.
* The fourth character should be a vowel

Sample Input:

A sentence that may/may not have valid Voter IDs

Sample Output:

Space separated list of valid voter IDs as given in test cases.

If no Valid voter ID is present in the sentence, print "NIL"

**TEST CASE 1:**

**INPUT:**

hjfRRDFU435681P KJDP898937JH HUHA4356365\*$(

**OUTPUT:**

RDFU435681 HUHA435636

**TEST CASE 2:**

**INPUT:**

hjhjgGGFSFE$#5435@#$ FGRE432451RREO340345

**OUTPUT:**

FGRE432451

**TEST CASE 3:**

**INPUT:**

43HOFKE435 GFGG3639453

**OUTPUT:**

NIL

Ans:

***import re***

***a=input()***

***pattern="[A-Z]{3}[AEIOU]{1}[1-8]{6}"***

***lst=re.findall(pattern,a)***

***if len(lst)==0:***

***print("NIL")***

***else:***

***for i in lst:***

***print(i,end=" ")***

**Q2.**

Given two numbers A and B as input, write a program to print twin primes within the range [A,B]. If two consecutive odd numbers are both prime then they are known as twin primes. The output should be a **list of tuples**. Print "NIL" if there are no twin primes in the given range. Also print the sum of numbers in each pair as a list.

Note: Your program should include a function to check and return if a number passed as argument is prime or not. Use this function at appropriate places in the program.

e.g., if A = 4 and B = 20; the list of twin primes' tuples are [(5,7),(11,13),(17,19)]. Here, 5 and 7 are two consecutive odd numbers, and both are prime.

Sample Input:

Two numbers A and B

Sample Output:

Line 1: List of twin prime tuples

Line 2: List of the Sum of each twin prime pair

**TEST CASE 1:**

**INPUT:**

500 650

**OUTPUT:**

[(521, 523), (569, 571), (599, 601), (617, 619), (641, 643)]

[1044,1140,1200,1236,1284]

**TEST CASE 2:**

**INPUT:**

10 100

**OUTPUT:**

[(11, 13), (17, 19), (29, 31), (41, 43), (59, 61), (71, 73)]

[24,36,60,84,120,144]

**TEST CASE 3:**

**INPUT:**

75 100

**OUTPUT:**

NIL

Ans:

***def prime(a):***

***for i in range(2,a//2):***

***if a%i==0:***

***return False***

***return True***

***a=input()***

***b=a.split(" ")***

***c,d=b***

***c,d=int(c),int(d)***

***lst=[]***

***lsts=[]***

***for i in range(c,d+1):***

***if i%2!=0:***

***if prime(i):***

***if prime(i+2):***

***tup=(i,i+2)***

***lst.append(tup)***

***if len(lst)==0:***

***print("NIL")***

***else:***

***print(lst)***

***for i in lst:***

***sum=i[0]+i[1]***

***lsts.append(sum)***

***print(lsts)***

**Q3.**

You are given a sentence that contain some EPIC numbers. Write a program using **Regular expression** (MUST) to find and print all the valid EPIC card numbers in the sentence.

A valid EPIC card number is a 10-length alphanumeric word that is:

* A combination of **alphabets (A to Z)** and **digits (from 0 to 5) only.**
* allows Uppercase letters only.
* starts with an alphabet and ends with a digit.
* Only the first three characters are alphabets.
* The second character should be a vowel.

Sample Input:

A sentence that may/may not have valid EPICs

Sample Output:

Space separated list of valid EPICs as given in test cases.

If no valid EPIC is present in the sentence, print "NIL"

**TEST CASE 1:**

**INPUT:**

hjfRRIFU435680P KJDP898937JH HUH34350305\*$(

**OUTPUT:**

HUH3435030

**TEST CASE 2:**

**INPUT:**

 hjhjgGGFSFE$#5435@#$ FUR0432450REO3403451984

**OUTPUT:**

 FUR0432450 REO3403451

**TEST CASE 3:**

**INPUT:**

 43HOFKE435 GTRGR539856

**OUTPUT:**

NIL

Ans:

***import re***

***pattern="[A-Z][AEIOU][A-Z][0-5]{7}"***

***n=input()***

***k=re.findall(pattern,n)***

***if k==[]:***

***print("NIL")***

***else:***

***for i in k:***

***print(i,end=" ")***

**Q6. (6 Marks)**

Given two numbers X and Y as input, write a program to print twin primes within the range [X,Y]. If two consecutive odd numbers are both prime then they are known as twin primes. The output should be a **list of tuples**. Print "NIL" if there are no twin primes in the given range. Also print the product of the numbers in each pair as a list.

Note: Your program should include a function to check and return if a number passed as argument is prime or not. Use this function at appropriate places in the program.

e.g., if X = 4 and Y = 20; the list of twin primes' tuples are [(5,7),(11,13),(17,19)]. Here, 5 and 7 are two consecutive odd numbers, and both are prime.

Sample Input:

Two numbers X and Y

Sample Output:

Line 1: List of twin prime tuples

Line 2: List of the product of each twin prime pair

**TEST CASE 1:**

**INPUT:**

75 100

**OUTPUT:**

NIL

**TEST CASE 2:**

**INPUT:**

300 400

**OUTPUT:**

[(311, 313), (347, 349)]

[97343,121103]

**TEST CASE 3:**

**INPUT:**

25 150

**OUTPUT:**

[(29, 31), (41, 43), (59, 61), (71, 73), (101, 103), (107, 109), (137, 139)]

[899,1763,3599,5183,10403,11663,19043]

Ans:

***def prime(n):***

***for i in range(2,n):***

***if n%i==0:***

***return False***

***else:***

***continue***

***return True***

***n=input().split()***

***l=[]***

***l1=[]***

***for i in range(int(n[0]),int(n[1])-1):***

***if prime(i) and prime(i+2):***

***l.append((i,i+2))***

***l1.append(i\*(i+2))***

***if l==[]:***

***print("NIL")***

***else:***

***print(l)***

***print(l1)***

**Q5.**

You are given a sentence that contain some PAN card numbers. Write a program using **Regular expression** (MUST) to find and print all the valid PAN card numbers in the sentence.

A valid PAN card number is a 10-length alphanumeric word that is:

* A combination of **alphabets (A to Z)** and **digits (from 3 to 9) only.**
* allows Uppercase letters only.
* starts with an alphabet and ends with a digit.
* Only the first five characters are alphabets.
* The first character should be a vowel.

Sample Input:

A sentence that may/may not have valid PAN IDs

Sample Output:

Space separated list of valid PAN IDs as given in test cases.

If no valid PAN ID is present in the sentence, print "NIL"

**TEST CASE 1:**

**INPUT:**

hjfURIFU435680P KIJDPM898937JH HUH34350305\*$(

**OUTPUT:**

URIFU43568 IJDPM89893

**TEST CASE 2:**

**INPUT:**

 hjhjgGGFSFE$#5435@#$ URBGO0432450AREOO3493451984

**OUTPUT:**

AREOO34934

**TEST CASE 3:**

**INPUT:**

 43HOFKE435 HYHRJ08656654

**OUTPUT:**

NIL

Ans:

***import re***

***s=input()***

***c=0***

***a=s.split(" ")***

***for i in a:***

***for j in range(0,len(i)):***

***k=j+10***

***if k>len(i):***

***break***

***s1=i[j:k]***

***pattern=re.compile("[AEIOU][A-Z]{4}[3-9]{5}")***

***b=re.search(pattern,s1)***

***if b!=None:***

***c=c+1***

***print(s1,end=" ")***

***if c==0:***

***print("NIL")***

**Q6.**

Given two numbers P and Q as input, write a program to print twin primes within the range [P,Q]. If two consecutive odd numbers are both prime then they are known as twin primes. The output should be a **list of tuples**. Print "NIL" if there are no twin primes in the given range. Also print the squares of the second number in each pair as a list.

Note: Your program should include a function to check and return if a number passed as argument is prime or not. Use this function at appropriate places in the program.

e.g., if P = 4 and Q = 20; the list of twin primes' tuples are [(5,7),(11,13),(17,19)]. Here, 5 and 7 are two consecutive odd numbers, and both are prime.

Sample Input:

Two numbers P and Q

Sample Output:

Line 1: List of twin prime tuples

Line 2: List of the squares of the second number in each pair

**TEST CASE 1:**

**INPUT:**

300 400

**OUTPUT:**

[(311, 313), (347, 349)]

[97969,121801]

**TEST CASE 2:**

**INPUT:**

100 200

**OUTPUT:**

[(101, 103), (107, 109), (137, 139), (149, 151), (179, 181), (191, 193), (197, 199)]

[10609, 11881, 19321, 22801, 32761, 37249, 39601]

**TEST CASE 3:**

**INPUT:**

350 410

**OUTPUT:**

NIL

Ans:

***def prime(n):***

***for i in range(2,int(n\*\*(1/2)+1)):***

***if n % i == 0:***

***return False***

***else:***

***return True***

***l = input().split()***

***p, q = int(l[0]), int(l[1])***

***if p%2 == 0:***

***p += 1***

***if q%2 == 0:***

***q -= 1***

***L = []***

***L2 = []***

***num = p***

***while num <= q:***

***if prime(num):***

***if prime(num + 2):***

***L.append((num,num+2))***

***num += 2***

***num += 2***

***if L:***

***print(L)***

***for i in L:***

***L2.append(i[1]\*\*2)***

***print(L2)***

***else:***

***print("NIL")***