Assignment Sheet
EVEN Semester 2021
B.Tech CSE/IT 6th Semester

Artificial Intelligence Lab (15B17CI574)

Instructions:

- Students have to do a mini project apart from the Lab Assignments.
- The evaluative lab assignments must be evaluated as per the given deadline. The total weightage of all day to day work is 60 Marks.
- There will be two lab tests of 20 marks each. Absence in Lab Test-2 means Fail in the lab course.
- All students are required to attend at least 80% labs. 15 marks are reserved for attendance.
- The evaluative lab assignments must be evaluated as per the given deadline from time to time.

Week-2

January 25-30, 2021

PRACTICE:

1. Record representation

Let us take an example; a student can have an enrollment number, name, branch, batch etc. What are the different ways to manage record of a student and how can we iterate the records of a student. There are different ways of record management. Let us discuss some of them one by one:

a. Lists:

```
>>> a=['text',23,'alpha25']
>>> a
>>> a[0]
>>> a[1]
>>> a[2]
>>> a[3]
>>> student=["ajay kumar",21,"BTECH","CSE"]
>>> student[0].split()[-1]
>>> student[0].split()[0]
>>>len(student)
>>> student[0].upper()
>>> student[0].swapcase()  //convert lower to upper and vice versa
>>> student[0].ljust(100)    //ljust,rjust,center are used for text alignment
>>> student[0].rjust(100)
>>> student[0].center(100)
>>> student[0].zfill(100)
>>> student[0].replace('ajay','vijay')
>>> student
```

b. Database List:

Database is group of lists.

```
>>>list1=['student1',21,'batch1']
>>> list2=['student2',22,'batch2']
```

```
>>> list1
>>> list2
>>> database=[list1,list2]
>>> for data in database:print data
>>> database[1][0]
>>> database[1][2]
>>> database[0][1]
>>> age=[data[1] for data in database]>>>age
>>> pays=map((lambda x:x[0]),database)
>>> pays
>>>database.append(['student3',23,'batch3'])
>>>database
```

c. Field Labels

```
>>>NAME,  AGE, PAY = range(3)              # [0, 1, 2]
>>>bob = ['Bob Smith', 42, 10000]
>>>bob[NAME]
>>>PAY, bob[PAY]
```

d. Dictionaries

Using list based dictionaries; you can attach values to the field names.

```
>>> list1={'name': 'student1','age': 21,'batch': 'batch1'}
>>> list2={'name': 'student2','age': 22,'batch': 'batch2'}
>>> list1['name']
>>> list1['age']+=1
```

2. File Handling

Writing data to a file

```
>>> file=open('data.txt','w')
>>>file.write('Hello file world!\n')
>>>file.write('Bye file world.\n')
>>>file.close()
```

Reading data from a file

```
>>> file=open('data.txt','r')
>>> for line in file.readlines():
        print line
>>>file.seek(0)
>>>file.read()
>>>file.seek(0)
>>>file.readlines()
>>>file.readline()
>>>file.seek(0)
>>>file.readline()
>>>file.seek(0)
```

```
>>>file.read(1),file.read(8)
>>>file.next()
>>>file.next()
>>>file.next()
```

EXERCISES

Q1. Write a program to compute the number of characters, words and lines in a file.

Q2. -Write function to compute gcd, lcm of two numbers.

Q3. -Write a program to implement Merge sort. Write a program to implement Selection sort, Insertion sort

Q4. Find the sum of all the primes below two million. Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89,...

Q5. By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

Q6. Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file?

Q7. Write a function ball_collide that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding.

Hint: Represent a ball on a plane as a tuple of (x, y, r), r being the radius If (distance between two balls centers) <= (sum of their radii) then (they are colliding)

Q8. Write a function nearly_equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.

Q9. Write a program to perform multiplication of two square matrices

Q10. Implement Stack in python.

Q11. Implement Queue in Python.

Q12. Implement Tree in Python.