Machine Learning Lab Assignment - 1

Fayaz Ahmed Shaik

AP20110010149

CSE - C

7. A simple program

```
from math import *
d = 10.0 # diameter
A = pi * d**2 / 4
print ("diameter =", d)
print ("area = ", A)
```

```
diameter = 10.0
area =  78.53981633974483
```

In the Above code, we have imported math function and found the area of the circle with diameter

8. User input

```
s = input("What is your name? ")
print ("HELLO ", s)
```

```
What is your name? Fayaz Ahmed
HELLO  Fayaz Ahmed
```

In this code we have used the input function to take input from the user and print the given input in the output.

10. Input with data conversion

```
x = int(input("Input an integer: "))
y = float(input("Input a float: "))
print (x, y)
```

```
Input an integer: 15
Input a float: 150.2
15 150.2
```

Using this code, we have took the input from the user in the form of integer and float value and printed the values.

```
from math import *
d = float(input("Diameter: "))
A = pi * d**2 / 4
print("Area = ", A)
```

```
Diameter: 24
Area =  452.3893421169302
```

It is similar to the code above, but in this code we took the input from the user in the form of float value and found the area of the circle.

11. While loops

```
from math import *

i = 0
while i<= 100:
  print (i, "\t\t" , sqrt(i))
  i = i + 1
print ("READY!")
```

```
    0               0.0
    1               1.0
    2               1.4142135623730951
    3               1.7320508075688772
    4               2.0
    5               2.23606797749979
    6               2.449489742783178
    7               2.6457513110645907
    8               2.8284271247461903
    9               3.0
    10              3.1622776601683795
    11              3.3166247903554
    12              3.4641016151377544
    13              3.605551275463989
    14              3.7416573867739413
    15              3.872983346207417
    16              4.0
    17              4.123105625617661
    18              4.242640687119285
    19              4.358898943540674
    20              4.47213595499958
    21              4.58257569495584
    22              4.69041575982343
    23              4.795831523312719
    24              4.898979485566356
    25              5.0
    26              5.0990195135927845
    27              5.196152422706632
    28              5.291502622129181
    29              5.385164807134504
    30              5.477225575051661
    31              5.5677643628300215
    32              5.656854249492381
    33              5.744562646538029
    34              5.830951894845301
    35              5.916079783099616
    36              6.0
    37              6.082762530298219
    38              6.164414002968976
    39              6.244997998398398
    40              6.324555320336759
    41              6.4031242374328485
    42              6.48074069840786
    43              6.557438524302
    44              6.6332495807108
    45              6.708203932499369
    46              6.782329983125268
    47              6.855654600401044
    48              6.928203230275509
    49              7.0
    50              7.0710678118654755
    51              7.14142842854285
    52              7.211102550927978
    53              7.280109889280518
    54              7.3484692283495345
    55              7.416198487095663
    56              7.483314773547883
    57              7.54983443527075
```

In this code, we have imported math function to use sqrt(squareroot) function and we run a loop using while loop and printed all the squareroots of values from 0 to 100.

12. Testing conditions: if, elif, else

```
s = input("Input your name: ")
if s == "Fayaz":
  print ("Hello ", s)
else:
  print ("Hello unknown")
```

```
    Input your name: Fayaz
    Hello  Fayaz
```

In this code, we have took input from the user and checked if the input matches with the pre-defined name,if matched then it prints as Hello <...> or else it prints Hello Unknown. We used if and else statements here.

```
s =input("Input your name: ")
if s == "Tom":
  print ("Hello ", s)
```

```
elif s == "Carmen":
  print ("I'm so glad to see you ", s)
elif s == "Sonia":
  print ("I didn't expect you ",s)
else:
  print ("Hello unknown")
```

```
    Input your name: Sonia
    I didn't expect you  Sonia
```

In this code we have used multiple if, ifelse, else statement to make decisions based on the input we have provided.

13. Tuples

```
x,y = 5, 3
coordinates = x,y
print (coordinates)

dimensions = (8, 5.0, 3.14)
print (dimensions)
print (dimensions[0])
print (dimensions[1])
print (dimensions[2])
```

```
    (5, 3)
    (8, 5.0, 3.14)
    8
    5.0
    3.14
```

Here, we have assigned some value to the variables and and printed the values the values based on the indexing.

14. Lists (arrays)

```
a=[0,1,2]
print(a)
a.append(5)
a.append( "Zapzoo")
print(a)

x=[] #creating an empty list
```

```
    [0, 1, 2]
    [0, 1, 2, 5, 'Zapzoo']
```

Here, we have used append function to insert the values at the end of the list.

```
mylist = ["black", "red", "orange"]
print(mylist[0])
print(mylist[1])
print(mylist[2])
```

```
    black
    red
    orange
```

It is simlar to the above example, by using the index in a list, we are printing a required outputs

15. Range: producing lists of integer numbers

```
r1 = range(11)            # 0...10
print(r1)                 # [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
r2 = range(5,16)          # 5...15
print(r2)                 # [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
r3 = range(4,21,2)        # 4...20 step 2
print(r3)                 # [4, 6, 8, 10, 12, 14, 16, 18, 20]
r4 = range(15, 4, -5)     # 15....5 step -5
print(r4)
```

```
    range(0, 11)
    range(5, 16)
```

```
     range(4, 21, 2)
     range(15, 4, -5)
```

Here, in the range function, it prints the minimum and maximum values inside the paranthesis and the last element will be stepsize(which skips specific number of terms in the range and print the remaining values in that specific range)

16. Producing lists of floating point numbers

```
import numpy as np
r5 = np.linspace(0,2,9)  #linspace ( <startvalue>, <stopvalue>, <number_of_values> )
print(r5)

r6 = np.logspace(2, 3, 9)
print(r6)

     [0.   0.25 0.5  0.75 1.   1.25 1.5  1.75 2.  ]
     [ 100.          133.35214322  177.827941    237.13737057  316.22776602
       421.69650343  562.34132519  749.89420933 1000.        ]
```

In the above code, we have imported numpy and used linspace function for floating points. in the syntax of the linspace(a,b,c) , here 'a' represents the starting value and 'b' represents the final value and 'c' represents the total no of values. Similarly logspace give logarithemic values (here 2 means 10 power 2 and 3 means 10 power 3).

17. Iterating through a list: the for loop

```
mynames = [ "Sam", "Pit", "Misch", "Fayaz" ]

for n in mynames:
  print("HELLO ", n)

     HELLO  Sam
     HELLO  Pit
     HELLO  Misch
     HELLO  Fayaz
```

In the above code, we have run a loop to print each of the elements of the list seperately.

18. Iterating with indexing

```
colours = [ "black", "brown", "red", "orange", "yellow", "green", "blue", "violet", "grey","white" ]
cv = list (enumerate (colours))

for c in cv:
  print(c[0], "\t", c[1])

     0        black
     1        brown
     2        red
     3        orange
     4        yellow
     5        green
     6        blue
     7        violet
     8        grey
     9        white
```

When we've to iterate the list and also have the access to the index, here the enumerate function gives the list of tuples cv, therfore it prints likeindex wise

19. Functions

```
def area(b, h):
# calculate area of a rectangle
  A = b * h
  return A
```

```
def perimeter(b, h):
#calulates perimeter of a rectangle
  P = 2 * (b+h)
  return P

# main program using defined functions
width = 5
height = 3
print("Area = ", area(width, height))
print("Perimeter = ", perimeter(width, height))

def greeting():
  print("HELLO")
# main program using defined functions
greeting()
```

```
Area =  15
Perimeter =  16
HELLO
```

Here, we have defined functions like area and perimeter seperately and used that function in the main program to solve the problems easily.

20. Avoiding for loops: vector functions

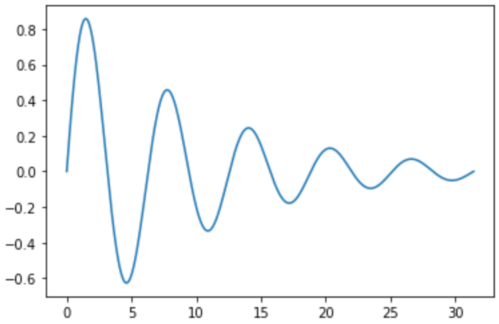
```
import numpy as np
# calculate 100 values for x and y without a for loop
x = np.linspace(0, 2* np.pi, 100)
y = np.sin(x)
print(x)
print(y)
```

```
[0.         0.06346652 0.12693304 0.19039955 0.25386607 0.31733259
 0.38079911 0.44426563 0.50773215 0.57119866 0.63466518 0.6981317
 0.76159822 0.82506474 0.88853126 0.95199777 1.01546429 1.07893081
 1.14239733 1.20586385 1.26933037 1.33279688 1.3962634  1.45972992
 1.52319644 1.58666296 1.65012947 1.71359599 1.77706251 1.84052903
 1.90399555 1.96746207 2.03092858 2.0943951  2.15786162 2.22132814
 2.28479466 2.34826118 2.41172769 2.47519421 2.53866073 2.60212725
 2.66559377 2.72906028 2.7925268  2.85599332 2.91945984 2.98292636
 3.04639288 3.10985939 3.17332591 3.23679243 3.30025895 3.36372547
 3.42719199 3.4906585  3.55412502 3.61759154 3.68105806 3.74452458
 3.8079911  3.87145761 3.93492413 3.99839065 4.06185717 4.12532369
 4.1887902  4.25225672 4.31572324 4.37918976 4.44265628 4.5061228
 4.56958931 4.63305583 4.69652235 4.75998887 4.82345539 4.88692191
 4.95038842 5.01385494 5.07732146 5.14078798 5.2042545  5.26772102
 5.33118753 5.39465405 5.45812057 5.52158709 5.58505361 5.64852012
 5.71198664 5.77545316 5.83891968 5.9023862  5.96585272 6.02931923
 6.09278575 6.15625227 6.21971879 6.28318531]
[ 0.00000000e+00  6.34239197e-02  1.26592454e-01  1.89251244e-01
  2.51147987e-01  3.12033446e-01  3.71662456e-01  4.29794912e-01
  4.86196736e-01  5.40640817e-01  5.92907929e-01  6.42787610e-01
  6.90079011e-01  7.34591709e-01  7.76146464e-01  8.14575952e-01
  8.49725430e-01  8.81453363e-01  9.09631995e-01  9.34147860e-01
  9.54902241e-01  9.71811568e-01  9.84807753e-01  9.93838464e-01
  9.98867339e-01  9.99874128e-01  9.96854776e-01  9.89821442e-01
  9.78802446e-01  9.63842159e-01  9.45000819e-01  9.22354294e-01
  8.95993774e-01  8.66025404e-01  8.32569855e-01  7.95761841e-01
  7.55749574e-01  7.12694171e-01  6.66769001e-01  6.18158986e-01
  5.67059864e-01  5.13677392e-01  4.58226522e-01  4.00930535e-01
  3.42020143e-01  2.81732557e-01  2.20310533e-01  1.58001396e-01
  9.50560433e-02  3.17279335e-02 -3.17279335e-02 -9.50560433e-02
 -1.58001396e-01 -2.20310533e-01 -2.81732557e-01 -3.42020143e-01
 -4.00930535e-01 -4.58226522e-01 -5.13677392e-01 -5.67059864e-01
 -6.18158986e-01 -6.66769001e-01 -7.12694171e-01 -7.55749574e-01
 -7.95761841e-01 -8.32569855e-01 -8.66025404e-01 -8.95993774e-01
 -9.22354294e-01 -9.45000819e-01 -9.63842159e-01 -9.78802446e-01
 -9.89821442e-01 -9.96854776e-01 -9.99874128e-01 -9.98867339e-01
 -9.93838464e-01 -9.84807753e-01 -9.71811568e-01 -9.54902241e-01
 -9.34147860e-01 -9.09631995e-01 -8.81453363e-01 -8.49725430e-01
 -8.14575952e-01 -7.76146464e-01 -7.34591709e-01 -6.90079011e-01
 -6.42787610e-01 -5.92907929e-01 -5.40640817e-01 -4.86196736e-01
 -4.29794912e-01 -3.71662456e-01 -3.12033446e-01 -2.51147987e-01
 -1.89251244e-01 -1.26592454e-01 -6.34239197e-02 -2.44929360e-16]
```

Here, we imported numpy and calculated vector function without using of the loops

21. Diagrams

```
from numpy import linspace, sin, exp, pi
import matplotlib.pyplot as mp
# calculate 500 values for x and y without a for loop
x = linspace(0, 10*pi, 500)
y = sin(x) * exp(-x/10)
# make diagram
mp.plot(x,y)
mp.show()
```



In the above code, we have imported matplotlib function which is used to display the graphical form of the output with the respective input type data.

23. Appendix

```
from numpy import *
print(sin(pi/4))
# With this import method the following would give an error:
#sin = 5 # naming conflict!
#print sin(pi/4)

import numpy as np
print(np.sin(np.pi/4))


from numpy import linspace, sin, exp, pi
print(sin(pi/4))
```

```
0.7071067811865475
0.7071067811865475
0.7071067811865475
```

Here we import numpy function which is a mathematical function and printing the value of sin(pi/4) in different formates.

✓ 0s completed at 10:55 PM