



Guru Gobind Singh Indraprastha University
University School of Automation & Robotics

Programming in Python (ICT 160)

Practical File

Name	Sujal Singh
Enrollment Number	04119051723
Batch	IOT-B1-23

Index

No.	Question	Remarks
1.	Write a program to perform string manipulation operations using set of pre-defined functions such as <code>find()</code> , <code>upper()</code> , <code>len()</code> , <code>max()</code> and <code>min()</code> , and fetch specific content from string.	
2.	Write a program to test and check the mathematical functions such as <code>ciel()</code> , <code>sqrt()</code> , <code>pow()</code> and <code>factorial()</code> .	
3.	Write a program that receives a number as input from user and returns if it's an odd or even number.	
4.	Write a program that receives input from the user to calculate the area of a triangle.	
5.	Write a program that receives input from the user to calculate the area of a square.	
6.	Write a program that receive input from the user to calculate the area of rectangle.	
7.	Write a program to check if the input string is a palindrome or not.	
8.	Write a program that receives marks of a student for a subject as input and assigns a grade A B C D E F.	
9.	Write a program to compute the GCD of two numbers.	
10.	Write a program to check if the given number is an Armstrong number or not. Examples of Armstrong number are 153, 370, 371 etc.	
11.	Write a program to check if the input year is a leap year or not	
12.	Write a program to compute factorial of a given number.	
13.	Write a Program to generate Fibonacci series till 100.	
14.	Write a program to print a multiplication table of a given number.	
15.	Write a program to create two lists and perform the following operation's: add the elements of the two list, compare the contents of the two list, find the number of elements in the list, sort the elements of the list and reverse the contents of the list.	
16.	Write a Program to create and display the content of the tuple. Initialize the tuple with the name of the cities. Display content of the tuple along with name/index positions of the cities.	
17.	Write a program to create an array of even numbers till 14. Display the contents of array, compute the length of the array and also show how to delete a element from the desired position from the array.	
18.	Using <code>filter()</code> function, write a program to filter the elements which are greater than 9.	
19.	Using filter function, write a program to display multiple of 5 from a given array.	

20.	Write a program to create a file called “input.txt”, perform write/read operation on it with a string “Computer Science”.	
21.	Write a program to create a file called “input.txt”, initialize it with a string of your choice and perform the read operation to read only the first 3 characters from the file.	
22.	Using NumPy, write a program to create 1-dimensional array, load it with numbers, and perform the operation of iteration and slicing on it.	
23.	Using NumPy, write a program to create multidimensional array, load it with the numbers and display the content of it.	
24.	Using NumPy, write a program to create two 1-dimensional array and perform the operation of iteration, sorting the contents of array and concatenating the contents of the array.	
25.	Using NumPy, initialize an array and display its dimensionality.	
26.	Using Pandas, create a dataframe, initialize it with the contents such as your enrollment Number and name and display them.	
27.	Create 2 arrays, using Matplotlib, plot the graph with the content of the two arrays, with coordinates plotted on x-axis and y-axis.	
28.	Create a .csv file (with contents like age, weight and BMI). Read the content of the file and using Pandas and Matplotlib, plot the graph.	
29.	Create a .csv file (with contents like age, weight and BMI). Read the content of the file and using Pandas and Matplotlib, plot the histogram.	
30.	Write a program to create a class called “Student” with fields such as: Enrollment Number, USS Name, Branch Name, Student Name, etc. Instantiate a class and make a call to user defined function to display the details of students.	
31.	Define Employee Class with fields such as Employee ID and Employee Name. Instantiate the class, invoke the constructor and make a call to user defined function to display the information about employee.	

1. Write a program to perform string manipulation operations using set of pre-defined functions such as:

- find()
- upper()
- len()
- max() and min()
- fetching specific content from string

Program:

```
1 string = "Hello, World!"
2
3 print(
4     f'string: "{string}"\n',
5     f'find("llo"): {string.find("llo")}',
6     f'upper(): {string.upper()}',
7     f'len(): {len(string)}',
8     f'max(): "{max(string)}"',
9     f'min(): "{min(string)}"',
10    f'fetching "llo": "{string[(start := string.find("llo")):start +
11        ↪ len("llo")]}"',
12    sep="\n"
```

Output:

```
1 string: "Hello, World!"
2
3 find("llo"): 2
4 upper(): HELLO, WORLD!
5 len(): 13
6 max(): "r"
7 min(): " "
8 fetching "llo": "llo"
```

2. Write a program to test and check the mathematical functions such as:

- ciel()
- sqrt()
- pow()
- factorial()

Program:	
1	<code>from math import ceil, sqrt, pow, factorial</code>
2	
3	<code>num = 3.14</code>
4	
5	<code>print(</code>
6	<code> f"num: {num}\n",</code>
7	<code> f"ceil(num): {ceil(num)}",</code>
8	<code> f"sqrt(num): {sqrt(num)}",</code>
9	<code> f"pow(num, 2): {pow(num, 2)}",</code>
10	<code> f"factorial(5): {factorial(5)}",</code>
11	<code> sep="\n"</code>
12	<code>)</code>
Output:	
1	num: 3.14
2	
3	ceil(num): 4
4	sqrt(num): 1.772004514666935
5	pow(num, 2): 9.8596
6	factorial(5): 120

3. Write a program that receives a number as input from user and returns if it's an odd or even number.

Program:	
1	<code>num = int(input("Enter an integer: "))</code>
2	<code>print(f"{num} is {'even' if num % 2 == 0 else 'odd'}.")</code>
Output:	
1	Enter an integer: 42
2	42 is even.

4. Write a program that receives input from the user to calculate the area of a triangle.

Program:	
1	<code>base = int(input("Enter base length of triangle(m): "))</code>
2	<code>height = int(input("Enter height of triangle(m): "))</code>
3	<code>print("area =", 0.5 * base * height)</code>
Output:	
1	Enter base length of triangle(m): 5
2	Enter height of triangle(m): 4
3	area = 10.0

5. Write a program that receives input from the user to calculate the area of a square.

Program:	
1	<code>side = int(input("Enter side length of square(m): "))</code>
2	<code>print(f"Area of square of side length {side} is {side * side}.")</code>
Output:	
1	Enter side length of square(m): 4
2	Area of square of side length 4 is 16.

6. Write a program that receives input from the user to calculate the area of a rectangle.

Program:	
1	<code>length = int(input("Enter length: "))</code>
2	<code>breadth = int(input("Enter breadth: "))</code>
3	<code>print(f"Area of rectangle having length {length} and breadth {breadth} is ↪ {length * breadth}.")</code>
Output:	
1	Enter length: 5
2	Enter breadth: 4
3	Area of rectangle having length 5 and breadth 4 is 20.

7. Write a program to check if the input string is a palindrome or not.

Program:	
1	<code>original = input("Enter a string: ")</code>
2	<code>print("Given string is", "a" if original == original[::-1] else "not a", ↪ "palindrome.")</code>
Output:	
1	Enter a string: ehehe
2	Given string is a palindrome.

8. Write a program that receives marks of a student for a subject as input and assigns a grade A||B||C||D||E||F.

Program:	
1	<code>offset = 9 - int(input("Enter marks for subject (0 - 100): ")) // 10</code>
2	<code># Makes negative offset values equal zero</code>
3	<code>positive_offset = (abs(offset) + offset) // 2</code>
4	<code>print(chr(65 + positive_offset) if offset <= 5 else "F")</code>
Output:	
1	Enter marks for subject (0 - 100): 90
2	A

9. Write a program to compute the GCD of two numbers.

Program:	
1	<code>def gcd(a, b):</code>
2	<code> # Using the Euclidean algorithm</code>
3	<code> if a < b:</code>
4	<code> return gcd(a, b - a)</code>
5	<code> elif a > b:</code>
6	<code> return gcd(a - b, b)</code>
7	<code> return a # if a equals b, then gcd is the same as a and b.</code>
8	
9	
10	<code>print(gcd(</code>
11	<code> int(input("Enter first number: ")),</code>
12	<code> int(input("Enter second number: "))</code>
13	<code>)</code>
Output:	
1	Enter first number: 10
2	Enter second number: 4
3	2

10. Write a program to check if the given number is an Armstrong number or not. Examples of Armstrong number are 153, 370, 371 etc.

Program:	
1	<code>num = int(str_num := input("Enter a number: "))</code>
2	
3	<code>power = len(str_num)</code>
4	<code>print(</code>
5	<code> "Given number is",</code>
6	<code> "not an" if sum([int(digit) ** power for digit in str_num]) != num else</code>
7	<code> "an",</code>
8	<code> "Armstrong number."</code>
8	<code>)</code>
Output:	
1	Enter a number: 153
2	Given number is an Armstrong number.

11. Write a program to check if the input year is a leap year or not.

Program:	
1	<code>from calendar import isleap</code>
2	
3	<code>print(</code>
4	<code> "Given year is",</code>
5	<code> "a" if isleap(int(input("Enter a year: "))) else "not a",</code>
6	<code> "leap year."</code>
7	<code>)</code>
Output:	
1	Enter a year: 2024
2	Given year is a leap year.

12. Write a program to compute factorial of a given number.

Program:	
1	<code>def fac(n):</code>
2	<code> if n == 0:</code>
3	<code> return 1</code>
4	<code> return n * fac(n - 1)</code>
5	
6	
7	<code>print(fac(int(input("Enter a number: "))))</code>
Output:	
1	Enter a number: 5
2	120

13. Write a program to generate the Fibonacci sequence till 100.

Program:	
1	<code>a, b = 0, 1</code>
2	<code>print(a, end=" ")</code>
3	<code>while b <= 100:</code>
4	<code> print(b, end=" ")</code>
5	<code> a, b = b, b + a</code>
Output:	
1	0 1 1 2 3 5 8 13 21 34 55 89

14. Write a program to print the multiplication table of a given number.

Program:	
1	<code>def table(n):</code>
2	<code> for i in range(1, 11):</code>
3	<code> print(f"{n} x {i} = {n * i}")</code>
4	
5	
6	<code>table(int(input("Enter a number: ")))</code>
Output:	
1	Enter a number: 3
2	3 x 1 = 3
3	3 x 2 = 6
4	3 x 3 = 9
5	3 x 4 = 12
6	3 x 5 = 15
7	3 x 6 = 18
8	3 x 7 = 21
9	3 x 8 = 24
10	3 x 9 = 27
11	3 x 10 = 30

15. Write a program to create two lists and perform the following operation's:

1. Add the elements of the two list.
2. Compare the contents of the two list.
3. Find the number of elements in the list.
4. Sort the elements of the list.
5. Reverse the contents of the list.

Program:	
1	<code>a, b = [1, 3, 2, 4, 5], [6, 5, 7, 9, 8]</code>
2	<code>print(</code>
3	<code> [x + y for x, y in zip(a, b)], # Only works for lists having same</code>
	<code> ↪ lengths</code>
4	<code> a[2] == a[3],</code>
5	<code> len(a), len(b),</code>
6	<code> sorted(b),</code>
7	<code> list(reversed(a)),</code>
8	<code> sep="\n"</code>
9	<code>)</code>

Output:	
1	[7, 8, 9, 13, 13]
2	False
3	5
4	5
5	[5, 6, 7, 8, 9]
6	[5, 4, 2, 3, 1]

16. Write a Program to create and display the content of the tuple. Initialize the tuple with the name of the cities. Display content of the tuple along with name/index positions of the cities.

Program:	
1	<code>cities = ("New Delhi", "Pune", "Jaipur", "Amritsar")</code>
2	<code>print(cities)</code>
3	<code>print(*[f"({i}): {name}" for i, name in enumerate(cities)], sep="\n")</code>
Output:	
1	<code>('New Delhi', 'Pune', 'Jaipur', 'Amritsar')</code>
2	<code>(0): New Delhi</code>
3	<code>(1): Pune</code>
4	<code>(2): Jaipur</code>
5	<code>(3): Amritsar</code>

17. Write a program to create an array of even numbers till 14. Display the contents of array, compute the length of the array and also show how to delete a element from the desired position from the array.

Program:	
1	<code>array = list(range(0, 15, 2))</code>
2	<code>print(array)</code>
3	<code>print(len(array))</code>
4	<code>del array[3]</code>
5	<code>print(array)</code>
Output:	
1	<code>[0, 2, 4, 6, 8, 10, 12, 14]</code>
2	<code>8</code>
3	<code>[0, 2, 4, 8, 10, 12, 14]</code>

18. Using `filter()` function, write a program to filter the elements which are greater than 9.

Program:	
1	<code>from random import randrange</code>
2	
3	<code>data = [randrange(0, 20) for _ in range(20)]</code>
4	<code>print(list(filter(lambda x: x > 9, data)))</code>

Output:

```
1 [12, 12, 17, 14, 15, 16, 12, 13, 11, 15, 14, 12]
```

19. Using `filter()` function, write a program to filter the elements which are greater than 9.

Program:

```
1 from random import randrange
2
3 data = [randrange(0, 100) for _ in range(30)]
4 print(data)
5 print(list(filter(lambda x: x % 5 == 0, data)))
```

Output:

```
1 [94, 39, 28, 55, 56, 39, 1, 53, 82, 24, 71, 20, 22, 36, 27, 40, 33, 67, 42,
   ↪ 94, 51, 41, 50, 3, 71, 15, 68, 29, 78, 79]
2 [55, 20, 40, 50, 15]
```

20. Write a program to create a file called “input.txt”, perform write/read operation on it with a string “Computer Science”.

Program:

```
1 with open("input.txt", "w") as file:
2     file.write("Computer Science")
3
4 with open("input.txt") as file:
5     print(file.read())
```

Output:

```
1 Computer Science
```

21. Write a program to create a file called “input.txt”, initialize it with a string of your choice and perform the read operation to read only the first 3 characters from the file.

Program:

```
1 with open("input.txt", "w") as file:
2     file.write("Hello, World!")
3
4 with open("input.txt") as file:
5     print(file.read(3))
```

Output:

```
1 Hel
```

22. Using NumPy, write a program to create 1-dimensional array, load it with numbers, and perform the operation of iteration and slicing on it.

Program:	
1	<code>import numpy as np</code>
2	
3	<code>print(arr := np.array([2, 3, 4, 1, 7, 6], np.int32))</code>
4	<code>for e in np.nditer(arr):</code>
5	<code> print(e, end=" ")</code>
6	<code>print()</code>
7	<code>print(arr[2:5])</code>

Output:	
1	<code>[2 3 4 1 7 6]</code>
2	<code>2 3 4 1 7 6</code>
3	<code>[4 1 7]</code>

23. Using NumPy, write a program to create multidimensional array, load it with the numbers and display the content of it.

Program:	
1	<code>import numpy as np</code>
2	
3	<code>arr = np.array([[1, 2, 3], [4, 5, 6]])</code>
4	<code>print(arr)</code>

Output:	
1	<code>[[1 2 3]</code>
2	<code> [4 5 6]]</code>

24. Using NumPy, write a program to create two 1-dimensional array and perform the operation of iteration, sorting the contents of array and concatenating the contents of the array.

Program:	
1	<code>import numpy as np</code>
2	
3	<code>arr = np.array([9, 4, 6, 2, 7])</code>
4	
5	<code>for e in np.nditer(arr):</code>
6	<code> print(e)</code>
7	
8	<code>print(np.sort(arr))</code>
9	<code>print(np.concatenate((arr, np.array([38, 77, 29, 84, 75])))</code>

Output:	
1	9
2	4
3	6
4	2
5	7
6	[2 4 6 7 9]
7	[9 4 6 2 7 38 77 29 84 75]

25. Using NumPy, initialize an array and display its dimensionality.

Program:	
1	<code>import numpy as np</code>
2	
3	<code>arr = np.array([[[[1, 2], [3, 4]], [[5, 6], [6, 7]], [[8, 9], [9, 10]]])</code>
4	<code>print(arr.shape)</code>
Output:	
1	(3, 2, 2)

26. Using Pandas, create a dataframe, initialize it with the contents such as your enrollment Number and name and display them.

Program:	
1	<code>import numpy as np</code>
2	<code>import pandas as pd</code>
3	
4	<code>class_list = pd.DataFrame(np.array([</code>
5	<code> ["Name", "Enrollment Number"],</code>
6	<code> ["Alice", 1],</code>
7	<code> ["Bob", 2],</code>
8	<code> ["Bob", 3]</code>
9	<code>]))</code>
10	
11	<code>print(class_list)</code>
Output:	
1	0 1
2	0 Name Enrollment Number
3	1 Alice 1
4	2 Bob 2
5	3 Bob 3

27. Create 2 arrays, using Matplotlib, plot the graph with the content of the two arrays, with coordinates plotted on x-axis and y-axis.

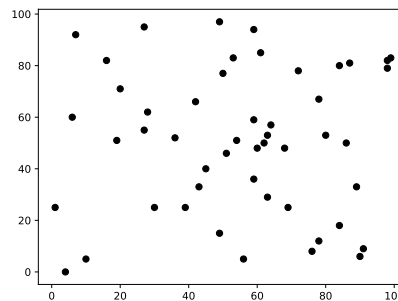
Program:

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 plt.plot(
5     np.array([np.random.randint(0, 100) for _ in range(50)]),
6     np.array([np.random.randint(0, 100) for _ in range(50)]),
7     "ko"
8 )
9 plt.show()

```

Output:



28. Create a .csv file (with contents like age, weight and BMI). Read the content of the file and using Pandas and Matplotlib, plot the graph.

data.csv			
Age	Height	Weight	BMI
21	185	109	31.84
47	173	74	4.72
46	174	78	5.76
34	171	66	2.57
33	174	76	5.10

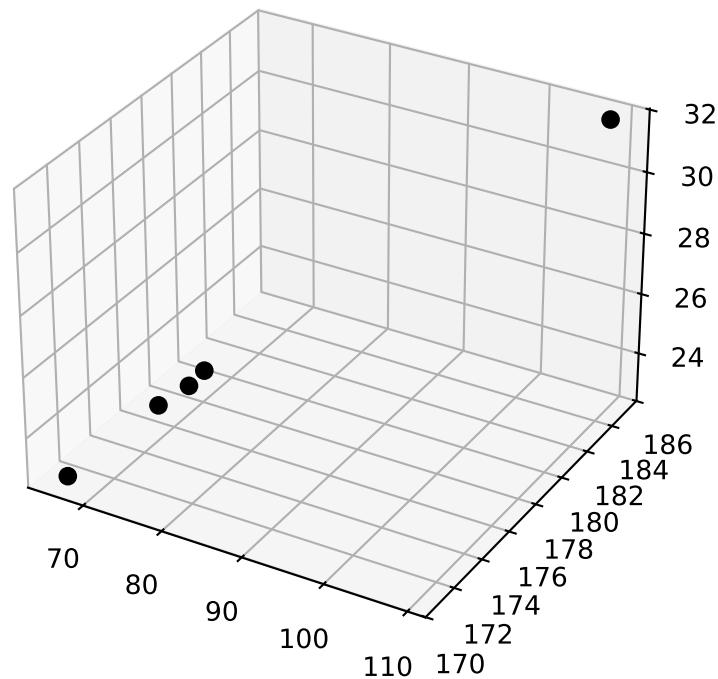
Program:

```

1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 data = pd.read_csv("data.csv")
5
6 fig, ax = plt.subplots(subplot_kw={"projection": "3d"})
7 ax.plot(data["Weight"], data["Height"], data["BMI"], "ko")
8
9 plt.show()

```

Output:

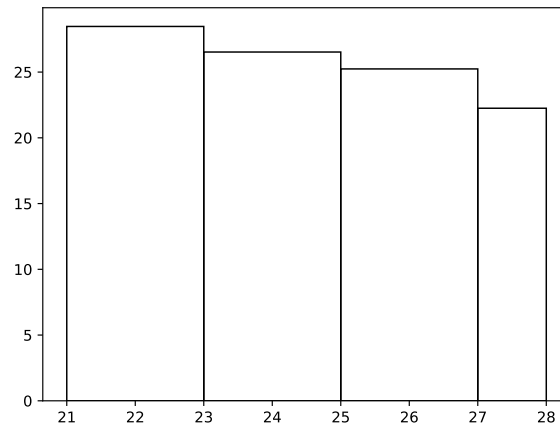


29. Create a .csv file (with contents like age, weight and BMI). Read the content of the file and using Pandas and Matplotlib, plot the histogram.

Program:

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 data = pd.read_csv("data.csv")
6
7 bmi = data["BMI"]
8 grp_size = 2
9 average_group_bmi = np.array([
10     sum(bmi[i:i + grp_size]) / grp_size
11     for i in range(0, len(bmi), grp_size)
12 ])
13
14 bins = data["Age"][0:len(data["Age"]):grp_size]
15 plt.hist(bins[:-1], bins, weights=average_group_bmi[:-1], fill=False)
16
17 plt.show()
```

Output:



30. Write a program to create a class called “Student” with fields such as: Enrollment Number, USS Name, Branch Name, Student Name, etc. Instantiate a class and make a call to user defined function to display the details of students.

Program:

```
1 class Student:
2     def __init__(self, enrollment_number, uss, branch, name):
3         self.enrollment_number = enrollment_number
4         self.uss = uss
5         self.branch = branch
6         self.name = name
7
8     def display(self):
9         print(
10             f"Name: {self.name}",
11             f"Enrollment Number: {self.enrollment_number}",
12             f"Branch: {self.branch}",
13             f"USS: {self.uss}",
14             sep="\n"
15         )
16
17
18 Student("04119051723", "USAR", "IIOT", "Sujal Singh").display()
```

Output:

```
1 Name: Sujal Singh
2 Enrollment Number: 04119051723
3 Branch: IIOT
4 USS: USAR
```


31. Write a program to create a class called "Student" with fields such as: Enrollment Number, USS Name, Branch Name, Student Name, etc. Instantiate a class and make a call to user defined function to display the details of students.

Program:

```
1 class Employee:
2     def __init__(self, emp_id, name):
3         self.emp_id = emp_id
4         self.name = name
5
6     def display(self):
7         print(f"Name: {self.name}\nID: {self.emp_id}")
8
9
10 Employee("04119051723", "Sujal Singh").display()
```

Output:

```
1 Name: Sujal Singh
2 ID: 04119051723
```