PYTHON-HACKERRANK

1.To print ‘Hello world”

print("Hello, World!")

2. Arithmetic operation:

if \_\_name\_\_ == '\_\_main\_\_':

a = int(input())

b = int(input())

print(a+b)

print(a-b)

print(a\*b)

3.Python division:

if \_\_name\_\_ == '\_\_main\_\_':

a = int(input())

b = int(input())

print(a//b)

print(a/b)

4. python loops:

if \_\_name\_\_ == '\_\_main\_\_':

n = int(input())

for i in range(n):

print(i\*\*2)

5. write a function leap year or not:

def is\_leap(year):

leap = False

if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):

return True

return leap

year = int(input())

print(is\_leap(year))

6. Print the list of integers from through as a string, without spaces:

if \_\_name\_\_ == '\_\_main\_\_':

n = int(input())

for i in range(1, n+1):

print(i, end="")

7. list-comprehensions

if \_\_name\_\_ == '\_\_main\_\_':

x = int(input())

y = int(input())

z = int(input())

n = int(input())

coordinates = [[i, j, k] for i in range(x + 1) for j in range(y + 1) for k in range(z + 1) if i + j + k != n]

print(coordinates)

8. find second maximum number in a list:

n = int(input()) # Number of participants

scores = list(map(int, input().split())) # List of scores

runner\_up = sorted(set(scores), reverse=True)[1] # Sorting unique scores and picking the second highest

print(runner\_up)

9. nested-list:

if \_\_name\_\_ == '\_\_main\_\_':

students = []

for \_ in range(int(input())):

name = input()

score = float(input())

students.append([name, score])

sorted scores = sorted(set(score for name, score in students)) second\_lowest = sorted\_scores[1]

second\_lowest\_students = sorted([name for name, score in students if score == second\_lowest])

for student in second\_lowest\_students:

print(student)

10. Finding the percentage:

if \_\_name\_\_ == '\_\_main\_\_':

n = int(input())

student\_marks = {}

for \_ in range(n):

name, \*line = input().split()

scores = list(map(float, line))

student\_marks[name] = scores

query\_name = input()# Read the number of entries

# Calculate the average of the marks

average\_marks = sum(student\_marks[query\_name]) / len(student\_marks[query\_name])

# Print the average formatted to 2 decimal places

print(f"{average\_marks:.2f}")