# **Name: Abdurrahman Qureshi**

# **Roll No: 242466**

Practical No: 6

**1) Write a python program to create a tuple having numbers till 20. Print half of its values in 1 line and another half in the next line.**

CODE:

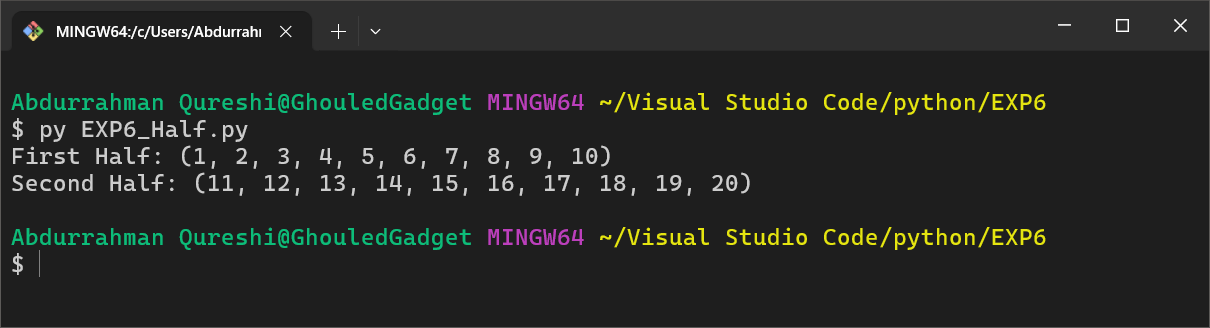
numbers = tuple(range(1, 21))

half = len(numbers) // 2

print("First Half:", numbers[:half])

print("Second Half:", numbers[half:])

OUTPUT:



**2) Write a python program to create a tuple having all numbers from 1 to 100. Now create another tuple whose values are even numbers in the first tuple.**

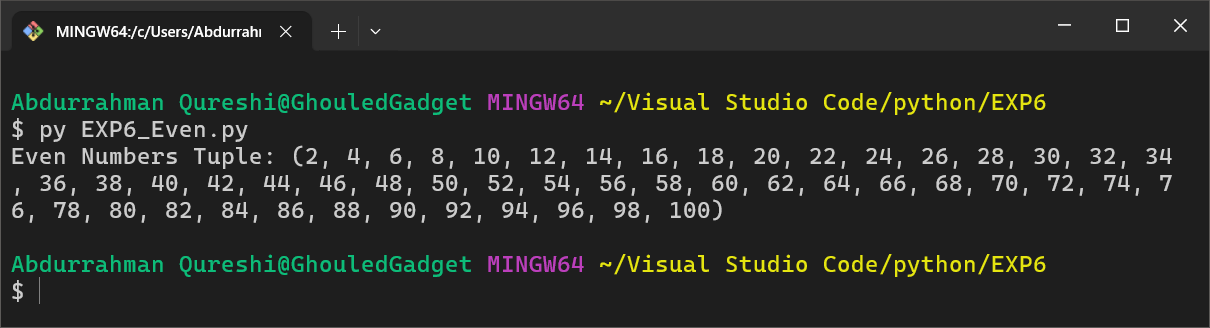
CODE:

numbers = tuple(range(1, 101))

even\_numbers = tuple(num for num in numbers if num % 2 == 0)

print("Even Numbers Tuple:", even\_numbers)

OUTPUT:



**3) Write a python program to concatenate two tuples.**

CODE:

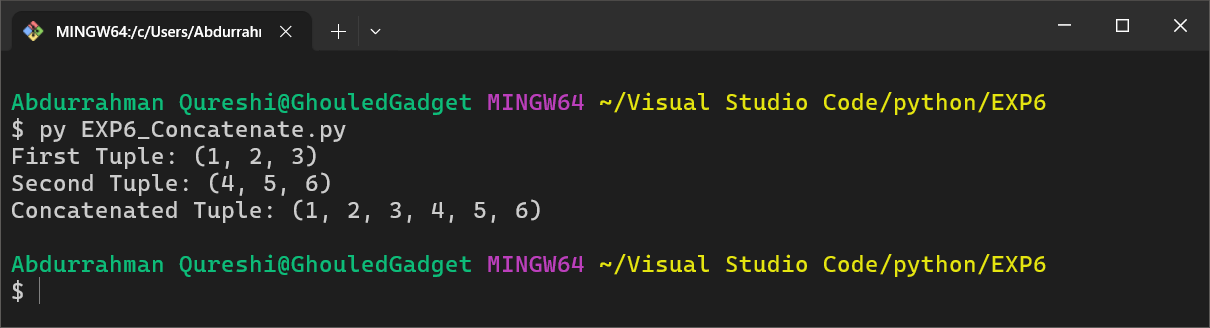
tuple1 = (1, 2, 3)

tuple2 = (4, 5, 6)

concatenated\_tuple = tuple1 + tuple2

print("Concatenated Tuple:", concatenated\_tuple)

OUTPUT:



**4) Write a python program to accept elements in the form of tuple and display their sum and average.**

CODE:

numbers = tuple(map(int, input("Enter numbers separated by space: ").split()))

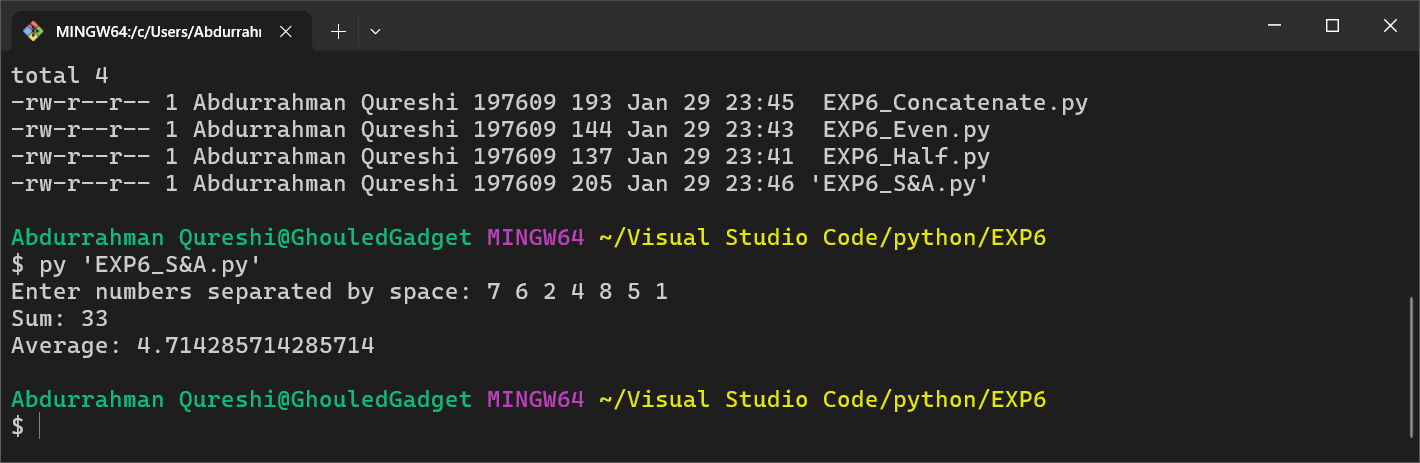
sum\_values = sum(numbers)

avg\_value = sum\_values / len(numbers)

print("Sum:", sum\_values)

print("Average:", avg\_value)

OUTPUT:



**5) Write a python program to find the first occurrence of an element in a tuple.**

CODE:

numbers = (10, 20, 30, 40, 50, 30, 60, 70)

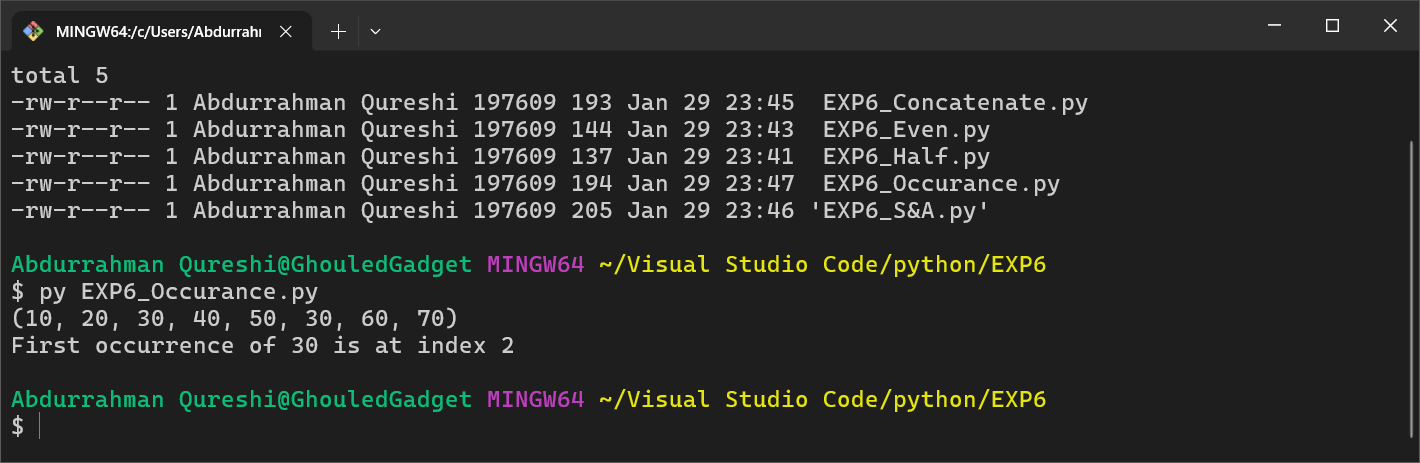
search\_element = 30

index = numbers.index(search\_element)

print(numbers)

print(f"First occurrence of {search\_element} is at index {index}")

OUTPUT:



**6) Write a python program to pickle List, dictionary, tuple and string.**

CODE:

import pickle

data = {

    "list": [1, 2, 3, 4, 5],

    "dict": {"name": "qarq90", "age": 20},

    "tuple": (10, 20, 30),

    "string": "Death is an old friend"

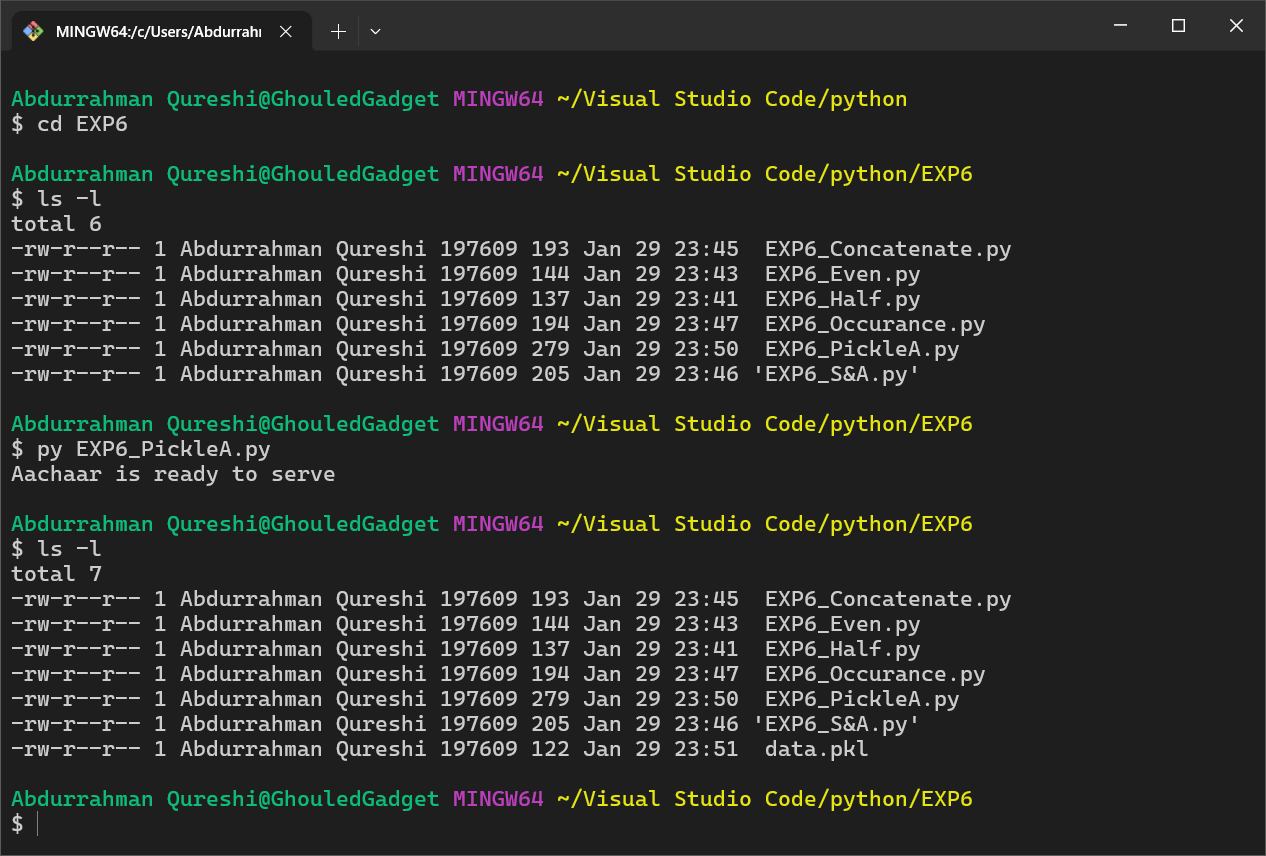
}

with open("data.pkl", "wb") as file:

    pickle.dump(data, file)

print("Aachaar is ready to serve")

OUTPUT:



**7) Write a python program to demonstrate 2 methods of the tuple.**

CODE:

numbers = (10, 20, 30, 40, 50, 30, 60, 30)

print(numbers)

print("Count of 30:", numbers.count(30))

print("Index of 40:", numbers.index(40))

OUTPUT:



**8) Write a python program to demonstrate 5 methods of set.**

CODE:

my\_set = {1, "2", 3.6, "Black People", "Hee hee haa"}

my\_set.add(6)

print("Set:", my\_set)

my\_set.remove(3.6)

print("Set:", my\_set)

my\_set.discard(500)

print("Set:", my\_set)

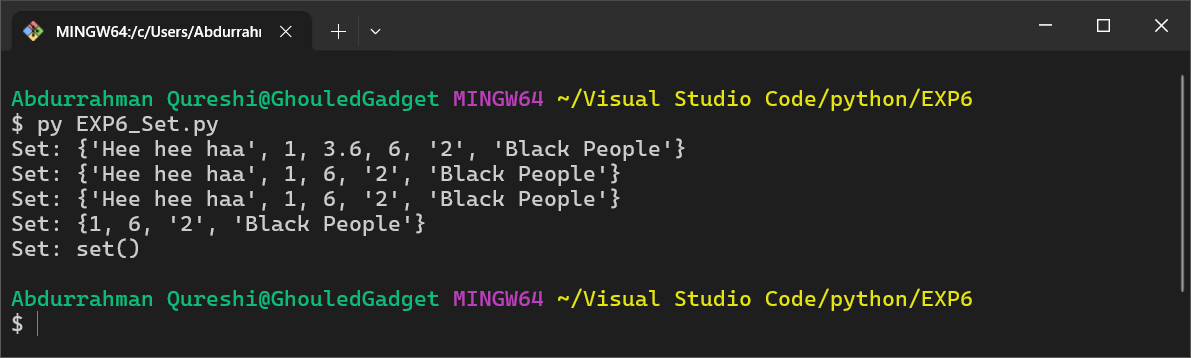
popped\_element = my\_set.pop()

print("Set:", my\_set)

my\_set.clear()

print("Set:", my\_set)

OUTPUT:



9) Write a python program to pickle tuples and strings.

CODE:

import pickle

data = {

    "tuple": (100, 200, 300),

    "string": "Aachar this string!"

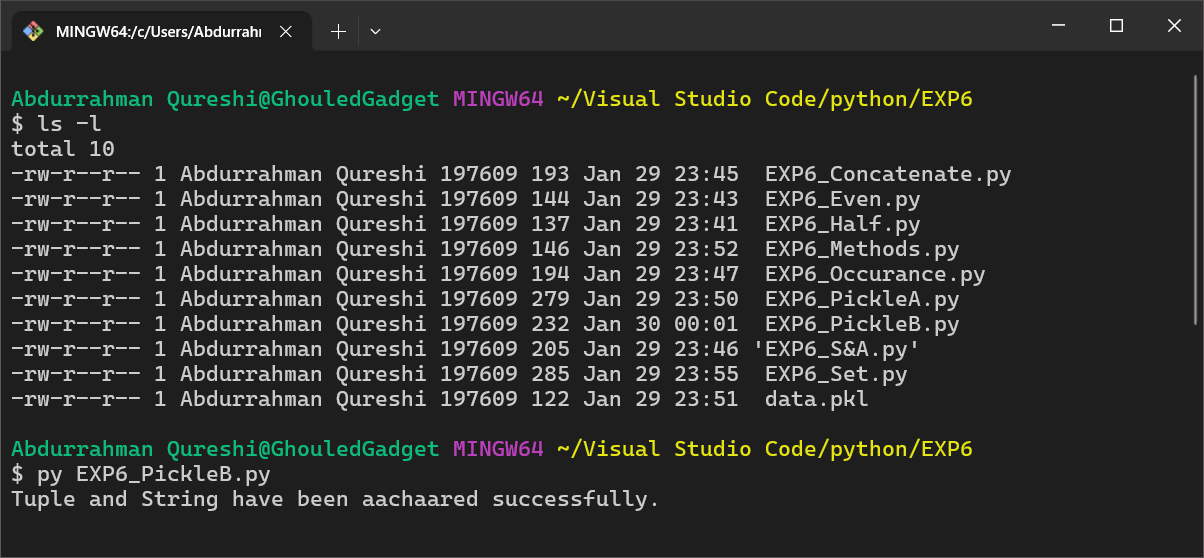
}

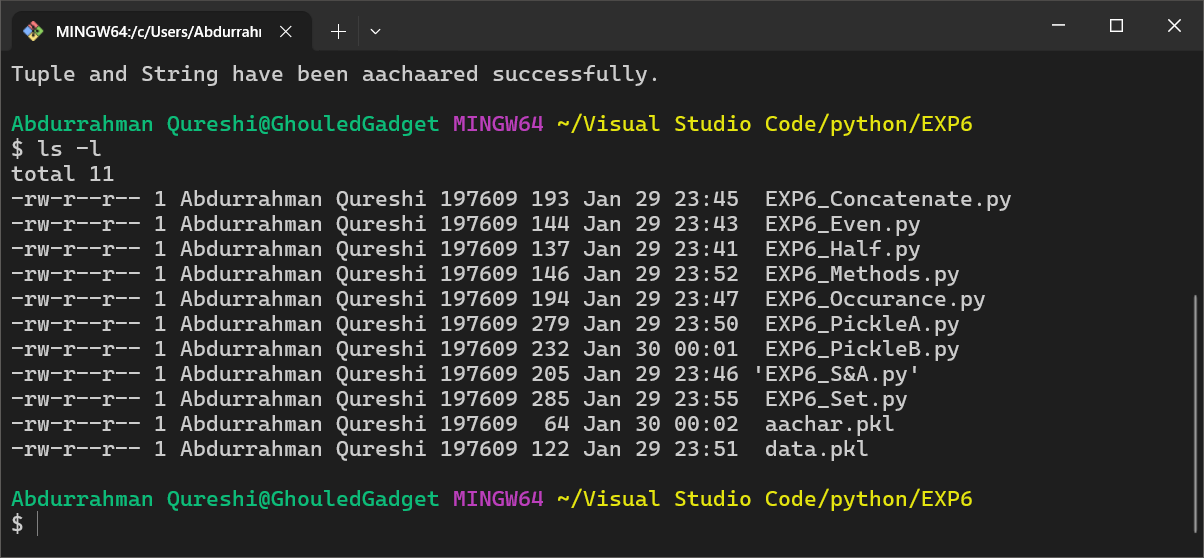
with open("aachar.pkl", "wb") as file:

    pickle.dump(data, file)

print("Tuple and String have been aachaared successfully.")

OUTPUT:





**10) Write a python program to implement snake ladder game using python dictionary**

CODE:

import time

import random

import sys

SLEEP\_BETWEEN\_ACTIONS = 1

MAX\_VAL = 100

DICE\_FACE = 6

snakes = {

17: 7,

54: 34,

62: 19,

64: 60,

87: 36,

95: 75,

93: 73,

98: 79,

}

ladders = {

4: 14,

9: 31,

1: 38,

21: 42,

28: 84,

51: 67,

72: 91,

80: 99

}

def welcome\_msg():

msg = "Welcome to Snake and Ladder Game"

print(msg)

def get\_player\_names():

player1\_name = input("Please enter a valid name for first player: ").strip()

player2\_name = input("Please enter a valid name for second player: ").strip()

print("\nMatch will be played between '" + player1\_name + "' and '" + player2\_name + "'\n")

return player1\_name, player2\_name

def get\_dice\_value():

time.sleep(SLEEP\_BETWEEN\_ACTIONS)

dice\_value = random.randint(1, DICE\_FACE)

print("Its a " + str(dice\_value))

return dice\_value

def got\_snake\_bite(old\_value, current\_value, player\_name):

print("\n" + "Snake Bite" + " ~~~~~~~~>")

print("\n" + player\_name + " got a snake bite. Down from " + str(old\_value) +" to " + str(current\_value))

def got\_ladder\_jump(old\_value, current\_value, player\_name):

print("\n" +"Woww , nailed it" + " ########")

print("\n" + player\_name + " climbed the ladder from " + str(old\_value) + " to " + str(current\_value))

def snake\_ladder(player\_name, current\_value, dice\_value):

time.sleep(SLEEP\_BETWEEN\_ACTIONS)

old\_value = current\_value

current\_value = current\_value + dice\_value

if current\_value > MAX\_VAL:

print("You need " + str(MAX\_VAL - old\_value) + " to win this game. Keep trying.")

return (old\_value)

print("\n" + player\_name + " moved from " + str(old\_value) + " to " + str(current\_value))

if current\_value in snakes:

final\_value = snakes.get(current\_value)

got\_snake\_bite(current\_value, final\_value, player\_name)

elif current\_value in ladders:

final\_value = ladders.get(current\_value)

got\_ladder\_jump(current\_value, final\_value, player\_name)

else:

final\_value = current\_value

return final\_value

def check\_win(player\_name, position):

time.sleep(SLEEP\_BETWEEN\_ACTIONS)

if MAX\_VAL == position:

print("\n\n\nThats it.\n\n" + player\_name + " won the game.")

print("Congratulations " + player\_name)

print("\nThank you for playing the game. Hope you enjoyed the game\n\n")

sys.exit(1)

welcome\_msg()

time.sleep(SLEEP\_BETWEEN\_ACTIONS)

player1\_name, player2\_name = get\_player\_names()

time.sleep(SLEEP\_BETWEEN\_ACTIONS)

player1\_current\_position = 0

player2\_current\_position = 0

while True:

time.sleep(SLEEP\_BETWEEN\_ACTIONS)

input\_1 = input("\n" + player1\_name + ": " + " Hit the enter to roll dice: ")

print("\nRolling dice...")

dice\_value = get\_dice\_value()

time.sleep(SLEEP\_BETWEEN\_ACTIONS)

print(player1\_name + " moving....")

player1\_current\_position = snake\_ladder(player1\_name, player1\_current\_position, dice\_value)

check\_win(player1\_name, player1\_current\_position)

input\_2 = input("\n" + player2\_name + ": " + " Hit the enter to roll dice: ")

print("\nRolling dice...")

dice\_value = get\_dice\_value()

time.sleep(SLEEP\_BETWEEN\_ACTIONS)

print(player2\_name + " moving....")

player2\_current\_position = snake\_ladder(player2\_name, player2\_current\_position, dice\_value)

check\_win(player2\_name, player2\_current\_position)

OUTPUT:

