Functions

Q1 :

Write a  Python Program to find the sum of the digits of the number recursively.

Q2 :

Write  a Python Program to find the binary equivalent of a number recursively.

Q3

Write a Python Program to check whether a given number is even or odd using betwise Ops

Q4:

This is a Python Program to find the GCD of two numbers using recursion.

Q5 :

This is a Python Program to find the product of two numbers using recursion.

Q6:

This is a Python Program to check whether a string is a palindrome or not using recursion.

Dictionary :

1. Which of the following statements create a dictionary?

a) d = {}

b) d = {“john”:40, “peter”:45}

c) d = {40:”john”, 45:”peter”}

d) All of the mentioned

2. What will be the output of the following Python code snippet?

```
1. d = {"john":40, "peter":45}
```

a) “john”, 40, 45, and “peter”

b) “john” and “peter”

c) 40 and 45

d) d = (40:”john”, 45:”peter”)

## 3. What will be the output of the following Python code snippet?

```
1. d = {"john":40, "peter":45}
2. "john" in d
```

a) True
b) False
c) None
d) Error

4. What will be the output of the following Python code snippet?

```
1. d1 = {"john":40, "peter":45}
2. d2 = {"john":466, "peter":45}
3. d1 == d2
```

a) True
b) False
c) None
d) Error

5. What will be the output of the following Python code snippet?

```
1. d1 = {"john":40, "peter":45}
2. d2 = {"john":466, "peter":45}
3. d1 > d2
```

a) True
b) False
c) Error
d) None

6. What will be the output of the following Python code snippet?

```
1. d = {"john":40, "peter":45}
2. d["john"]
```

a) 40
b) 45
c) “john”
d) “peter”

7. Suppose d = {“john”:40, “peter”:45}, to delete the entry for “john” what command do we use?
a) d.delete(“john”:40)
b) d.delete(“john”)
c) del d[“john”]
d) del d(“john”:40)

8. Suppose d = {“john”:40, “peter”:45}. To obtain the number of entries in dictionary which command do we use?
a) d.size()
b) len(d)
c) size(d)
d) d.len()

## 9. What will be the output of the following Python code snippet?

```
1. d = {"john":40, "peter":45}
2. print(list(d.keys()))
```

a) [“john”, “peter”]
b) [“john”:40, “peter”:45]
c) (“john”, “peter”)
d) (“john”:40, “peter”:45)

10. Suppose d = {“john”:40, “peter”:45}, what happens when we try to retrieve a value using the expression d[“susan”]?
a) Since “susan” is not a value in the set, Python raises a KeyError exception
b) It is executed fine and no exception is raised, and it returns None
c) Since “susan” is not a key in the set, Python raises a KeyError exception
d) Since “susan” is not a key in the set, Python raises a syntax error