

NAME

ADNAN

ID

13507

SUBJECT

MODERN PROGRAMMING LANGUAGES.

Q1.

(a).

Solved.

#Addition

```
>>>5+3
```

#Subtraction

```
>>>5-3
```

#Multiplication

```
>>>5*3
```

#Division

```
>>>5/3
```

```
Print (5+3)
```

```
Print (5-3)
```

```
Print (5*3)
```

```
Print (5/3)
```

Output:

8

2

15

(B)

Solved.

#Favorite Number

```
fav_ num = 18
```

```
msg = "My favorite number is" + str(fav_num) + "."
```

```
Print(msg)
```

Output:

```
My favorite number is 18
```

Q2.

(a)

Solved.

#Guests List

```
guests = ['Ali', 'Usman', 'Zeeshan']
```

```
name = guests [0] .title ()
```

```
Print (name + " , Please come to dinner .")
```

```
name = guests [1] .title ()
```

```
Print (name + " , Please come to dinner .")
```

```
name = guests [2] .title ()
```

```
Print (name + " , Please come to dinner .")
```

Continuous...

Output:

Ali , Please come to dinner.

Usman , Please come to dinner.

Zeeshan , Please come to dinner.

(b).

Solved.

```
#INVITE SOME PEOPLE TO DINNER
```

#Guests List

```
guests = ['Ali', 'Usman', 'Zeeshan']
```

```
name = guests [0] .title ()
```

```
Print (name + " , Please come to dinner .")
```

```
name = guests [1] .title ()
```

```
Print (name + " , Please come to dinner .")
```

```
name = guests [2] .title ()
```

```
Print (name + " , Please come to dinner .")
```

```
name = guests [1] , title ()
```

```
print ("\n sorry." + name + "can't make it to dinner.")
```

```
#Usman can't make it! Let's invite Bilal instead.
```

```
del (guests [1])
```

```
guests. Insert (1, ' Bilal')
```

```
#Print the invitations again.
```

```
name = guests [0] .title()
```

```
print ("\n" + name + " , Please come to dinner.")
```

```
name = guests [1] .title()
print (“\n” + name +”, Please come to dinner.”)
name = guests [2] .title()
print (“\n” + name +”, Please come to dinner.”)
```

Output:

Ali , Please come to dinner.

Usman , Please come to dinner.

Zeeshan , Please come to dinner.

Sorry, Usman can’t make it to dinner!

Output:

Ali , Please come to dinner.

Bilal , Please come to dinner.

Zeeshan , Please come to dinner.

Q3.

Solved.

```
Locations = ['Istanbul Turkey', 'Canada', 'Rome', 'Himalaya', 'Muree']
```

```
Print (“Original Order:”)
```

```
Print (locations)
```

```
Print (“\nAlphabetical :”)
```

```
Print (stored(locations))
```

```
Print (“\nOriginal order: “)
```

```
Print (locations)
Print ("\nReverse alphabetical :")
Print (stored (locations, reversed=True))
Print ("\nOriginal order: ")
Print (locations)
Print ("\nReversed :")
locations.reverse ()
Print (locations)
Print ("\nOriginal order: ")
locations.reverse ()
Print (locations)
Print ("\nAlphabetical :")
locations.sort ()
Print (locations)
Print ("\nReverse alphabetical :")
Locations. Sort (reverse=True)
Print (locations)
```

Output:

Original order:

```
['Istanbul Turkey', 'Canada', 'Rome', 'Himalaya', 'Muree']
```

Alphabetical order:

```
['Canada', 'Himalaya', 'Istanbul Turkey', 'Muree', 'Rome']
```

Original order:

['Istanbul Turkey', 'Canada', 'Rome', 'Himalaya', 'Muree']

Reverse Alphabetical order:

['Canada', 'Himalaya', 'Istanbul Turkey', 'Muree', 'Rome']

Original order:

['Istanbul Turkey', 'Canada', 'Rome', 'Himalaya', 'Muree']

Reverse order:

['Muree', 'Himalaya', 'Rome', 'Canada', 'Istanbul Turkey']

Original order:

['Istanbul Turkey', 'Canada', 'Rome', 'Himalaya', 'Muree']

Alphabetical order:

['Canada', 'Himalaya', 'Istanbul Turkey', 'Muree', 'Rome']

Reverse Alphabetical order:

['Canada', 'Himalaya', 'Istanbul Turkey', 'Muree', 'Rome']

----- **THE END** -----

