Assignmnet: 1 (Part A)

Question 1) Store the names of a few of your friends in a list called names. Print each person’s name by accessing each element in the list, one at a time.

In [ ]:

```
Name = ['Spandan','Bhaskar','Abhishek','Rajveer','Abhirub']
print(Name)

print(Name[0])
print(Name[1])
print(Name[2])
print(Name[3])
print(Name[4])
```

```
['Spandan', 'Bhaskar', 'Abhishek', 'Rajveer', 'Abhirub']
Spandan
Bhaskar
Abhishek
Rajveer
Abhirub
```

Question 2) Start with the list we used in Previous Exercise (Question 1), but instead of just printing each person’s name, print a message to them. The text of each message should be the same, but each message should be personalized with the person’s name.

In [ ]:

```
Name = ['Spandan','Bhaskar','Abhishek','Rajveer','Abhirub']
print(f"hello {Name[0]} have a good day")
print(f"hello {Name[1]} have a good day")
print(f"hello {Name[2]} have a good day")
print(f"hello {Name[3]} have a good day")
print(f"hello {Name[4]} have a good day")
```

```
hello Spandan have a good day
hello Bhaskar have a good day
hello Abhishek have a good day
hello Rajveer have a good day
hello Abhirub have a good day
```

Question 3) Think of your favorite mode of transportation, such as a motorcycle or a car, and make a list that stores several examples. Use your list to print a series of statements about these items, such as “I would like to own a Honda motorcycle.”

In [ ]:

```
Car = ['TataMotors','Audi','Thar','Jaguar', 'lamborgini']

print(f"I would like to own a  {Car[0]}")
print(f"I would like to own a {Car[1]}")
print(f"I would like to own a {Car[2]}")
print(f"I would like to own a {Car[3]}")
print(f"I would like to own a {Car[4]}")
```

```
I would like to own a  TataMotors
I would like to own a Audi
I would like to own a Thar
I would like to own a Jaguar
I would like to own a lamborgini
```

Assignmnet:1 (Part B)

Question 1) If we could invite anyone, living or deceased, to dinner, who would you invite? Make a list that includes at least three people you'd like to invite to dinner. Then use your list to print a message to each person, inviting them to dinner.

In [ ]:

```
Guests = ['Spandan','Bhaskar','Abhisekh']
print(Guests)
print(f"\n Hello!, {Guests[0]}.I am pleased to inform you to come at my home and ha
print(f"\n Hello!, {Guests[1]}.I am pleased to inform you to come at my home and ha
print(f"\n Hello!, {Guests[2]}.I am pleased to inform you to come at my home and ha
```

```
['Spandan', 'Bhaskar', 'Abhisekh']

 Hello!, Spandan.I am pleased to inform you to come at my home and have a dinner.
Thank you, will be waiting for you

 Hello!, Bhaskar.I am pleased to inform you to come at my home and have a dinner.
Thank you, will be waiting for you

 Hello!, Abhisekh.I am pleased to inform you to come at my home and have a dinner.
Thank you, will be waiting for you
```

Question 2) You just heard that one of your guests can't make the dinner, so you need to send out a new set of invitations. You'll have to think of someone else to invite.

• Start with your program from previous Excercise. Add a print() call at the end of your program, stating the name of the guest who can't make it.

• Modify your list, replacing the name of the guest who can't make it with the name of the new person you are inviting.

• Print a second set of invitation messages, one for each person who is still in your list.

In [ ]:

```
Guests = ['Spandan','Bhaskar','Abhisekh']
print(Guests)
Guest_cant_come = 'Bhaskar'
print(f"Guest who cant attend the dinner is {Guest_cant_come}, Hope he will attend

Guests[1] = 'Rajveer'
print(Guests)
Guest_Can_Come = 'Rajveer'
print(f"I am happy in inviting you {Guest_Can_Come} for the dinner in replacement o


print(Guests)
print(f"\n Hello!, {Guests[0]}.I am pleased to inform you to come at my home and ha
print(f"\n Hello!, {Guests[1]}.I am pleased to inform you to come at my home and ha
print(f"\n Hello!, {Guests[2]}.I am pleased to inform you to come at my home and ha
['Spandan', 'Bhaskar', 'Abhisekh']
Guest who cant attend the dinner is Bhaskar, Hope he will attend next time.
['Spandan', 'Rajveer', 'Abhisekh']
I am happy in inviting you Rajveer for the dinner in replacement of Bhaskar.Due to
some health issues Bhaskar couldn't attend the party. hope he will be fine.
['Spandan', 'Rajveer', 'Abhisekh']

 Hello!, Spandan.I am pleased to inform you to come at my home and have a dinner.
Thank you, will be waiting for you

 Hello!, Rajveer.I am pleased to inform you to come at my home and have a dinner.
Thank you, will be waiting for you

 Hello!, Abhisekh.I am pleased to inform you to come at my home and have a dinner.
Thank you, will be waiting for you
```

3-6. More Guests:

You just found a bigger dinner table, so now more space is available. Think of three more guests to invite to dinner.

• Start with your program from Exercise 3-4 or 3-5. Add a print() call to the end of your program, informing people that you found a bigger table.

• Use insert() to add one new guest to the beginning of your list.

• Use insert() to add one new guest to the middle of your list.

• Use append() to add one new guest to the end of your list.

• Print a new set of invitation messages, one for each person in your list.

In [ ]:

```python
Guests = ['Spandan', 'Rajveer', 'Abhisekh']
print(Guests)
print(f"\n Hello!, {Guests[0]}.We have found the bigger table")
print(f"\n Hello!, {Guests[1]}.We have found the bigger table")
print(f"\n Hello!, {Guests[2]}.We have found the bigger table\n")



Guests = ['Spandan', 'Rajveer', 'Abhisekh']
Guests.insert(0,'Bhaskar')
Guests.insert(3,'Sichitaj')
Guests.append('Depaak')
print(Guests)


print(Guests)
print(f"\n Hello!, {Guests[0]}.I am pleased to inform you to come at my home and ha
print(f"\n Hello!, {Guests[1]}.I am pleased to inform you to come at my home and ha
print(f"\n Hello!, {Guests[2]}.I am pleased to inform you to come at my home and ha
print(f"\n Hello!, {Guests[3]}.I am pleased to inform you to come at my home and ha
print(f"\n Hello!, {Guests[4]}.I am pleased to inform you to come at my home and ha
print(f"\n Hello!, {Guests[5]}.I am pleased to inform you to come at my home and ha
['Spandan', 'Rajveer', 'Abhisekh']

 Hello!, Spandan.We have found the bigger table

 Hello!, Rajveer.We have found the bigger table

 Hello!, Abhisekh.We have found the bigger table

['Bhaskar', 'Spandan', 'Rajveer', 'Sichitaj', 'Abhisekh', 'Depaak']
['Bhaskar', 'Spandan', 'Rajveer', 'Sichitaj', 'Abhisekh', 'Depaak']

 Hello!, Bhaskar.I am pleased to inform you to come at my home and have a dinner.
Thank you, will be waiting for you

 Hello!, Spandan.I am pleased to inform you to come at my home and have a dinner.
Thank you, will be waiting for you

 Hello!, Rajveer.I am pleased to inform you to come at my home and have a dinner.
Thank you, will be waiting for you

 Hello!, Sichitaj.I am pleased to inform you to come at my home and have a dinner.
Thank you, will be waiting for you

 Hello!, Abhisekh.I am pleased to inform you to come at my home and have a dinner.
Thank you, will be waiting for you

 Hello!, Depaak.I am pleased to inform you to come at my home and have a dinner. T
hank you, will be waiting for you
```

3-7. Shrinking Guest List:

You just found out that your new dinner table won't arrive in time for the dinner, and now you have space for only two guests.

• Start with your program from Exercise 3-6. Add a new line that prints a message saying that you can invite only two people for dinner.

• Use pop() to remove guests from your list one at a time until only two names remain in your list. Each time you pop a name from your list, print a message to that person letting them know you're sorry you can't invite them to dinner.

• Print a message to each of the two people still on your list, letting them know they're still invited.

• Use del to remove the last two names from your list, so you have an empty list. Print your list to make sure you actually have an empty list at the end of your program.

In [ ]:

```
Guests = ['Bhaskar', 'Spandan', 'Rajveer', 'Sichitaj', 'Abhisekh', 'Depaak']
print(Guests)
print(f"\n Hello!, {Guests[0]}.Sorry I can invite only two people")
print(f"\n Hello!, {Guests[1]}.Sorry I can invite only two people")
print(f"\n Hello!, {Guests[2]}.Sorry I can invite only two people")
print(f"\n Hello!, {Guests[3]}.Sorry I can invite only two people")
print(f"\n Hello!, {Guests[4]}.Sorry I can invite only two people")
print(f"\n Hello!, {Guests[5]}.Sorry I can invite only two people\n")



Guests = ['Bhaskar', 'Spandan', 'Rajveer', 'Sichitaj', 'Abhisekh', 'Depaak']
print(f"sorry i cannot invite you {Guests.pop()}")
print(f"sorry i cannot invite you {Guests.pop()}")
```

```python
print(f"sorry i cannot invite you {Guests.pop()}")
print(f"sorry i cannot invite you {Guests.pop()}\n")
print(Guests)
print(f"\nI am happy to you for the dinner {Guests[0]}.See you in the party")
print(f"\nI am happy to you for the dinner {Guests[1]}.See you in the party")


print(Guests)
del Guests[0]
del Guests[0]
print(Guests)
```

```
['Bhaskar', 'Spandan', 'Rajveer', 'Sichitaj', 'Abhisekh', 'Depaak']

 Hello!, Bhaskar.Sorry I can invite only two people

 Hello!, Spandan.Sorry I can invite only two people

 Hello!, Rajveer.Sorry I can invite only two people

 Hello!, Sichitaj.Sorry I can invite only two people

 Hello!, Abhisekh.Sorry I can invite only two people

 Hello!, Depaak.Sorry I can invite only two people

sorry i cannot invite you Depaak
sorry i cannot invite you Abhisekh
sorry i cannot invite you Sichitaj
sorry i cannot invite you Rajveer

['Bhaskar', 'Spandan']

I am happy to you for the dinner Bhaskar.See you in the party

I am happy to you for the dinner Spandan.See you in the party
['Bhaskar', 'Spandan']
[]
```

3-8. Seeing the World:

Think of at least five places in the world you'd like to visit.

• Store the locations in a list. Make sure the list is not in alphabetical order.

• Print your list in its original order. Don't worry about printing the list neatly; just print it as a raw Python list.

• Use sorted() to print your list in alphabetical order without modifying the actual list.

• Show that your list is still in its original order by printing it.

• Use sorted() to print your list in reverse-alphabetical order without changing the order of the original list.

• Show that your list is still in its original order by printing it again.

• Use reverse() to change the order of your list. Print the list to show that its order has changed.

- Use reverse() to change the order of your list again. Print the list to show it’s back to its original order.

- Use sort() to change your list so it’s stored in alphabetical order. Print the list to show that its order has been changed.

- Use sort() to change your list so it’s stored in reverse-alphabetical order. Print the list to show that its order has changed.

In [ ]:

```
Locations = ['Kolkata', 'Bhubaneshwar','Delhi','Ahemdabad']
print(f'\n1) {Locations}')
print(f'\n2) {sorted(Locations)}')
print(f'\n3) {Locations}')
print(f'\n4) {sorted(Locations,reverse=True)}')
Locations.reverse()
print(f'\n5) {(Locations)}')
Locations.reverse()
print(f'\n6) {(Locations)}')

Locations.sort()
print(f'\n7) {(Locations)}')
Locations.sort(reverse=True)
print(f'\n7) {(Locations)}')
```

```
1) ['Kolkata', 'Bhubaneshwar', 'Delhi', 'Ahemdabad']

2) ['Ahemdabad', 'Bhubaneshwar', 'Delhi', 'Kolkata']

3) ['Kolkata', 'Bhubaneshwar', 'Delhi', 'Ahemdabad']

4) ['Kolkata', 'Delhi', 'Bhubaneshwar', 'Ahemdabad']

5) ['Ahemdabad', 'Delhi', 'Bhubaneshwar', 'Kolkata']

6) ['Kolkata', 'Bhubaneshwar', 'Delhi', 'Ahemdabad']

7) ['Ahemdabad', 'Bhubaneshwar', 'Delhi', 'Kolkata']

7) ['Kolkata', 'Delhi', 'Bhubaneshwar', 'Ahemdabad']
```

3-9. Dinner Guests: Working with one of the programs from Exercises 3-4 through 3-7 (pages 41–42), use len() to print a message indicating the number of people you’re inviting to dinner.

In [ ]:

```
Guests = ['Bhaskar', 'Spandan', 'Rajveer', 'Sichitaj', 'Abhisekh', 'Depaak']
print(Guests)
print(f"\n Hello!, {Guests[0]}.Sorry I can invite only two people")
print(f"\n Hello!, {Guests[1]}.Sorry I can invite only two people")
print(f"\n Hello!, {Guests[2]}.Sorry I can invite only two people")
print(f"\n Hello!, {Guests[3]}.Sorry I can invite only two people")
print(f"\n Hello!, {Guests[4]}.Sorry I can invite only two people")
print(f"\n Hello!, {Guests[5]}.Sorry I can invite only two people\n")

print(len(Guests))
```

```
['Bhaskar', 'Spandan', 'Rajveer', 'Sichitaj', 'Abhisekh', 'Depaak']

 Hello!, Bhaskar.Sorry I can invite only two people

 Hello!, Spandan.Sorry I can invite only two people

 Hello!, Rajveer.Sorry I can invite only two people

 Hello!, Sichitaj.Sorry I can invite only two people

 Hello!, Abhisekh.Sorry I can invite only two people

 Hello!, Depaak.Sorry I can invite only two people

6
```

3-10. Every Function: Think of things you could store in a list. For example, you could make a list of mountains, rivers, countries, cities, languages, or anything else you'd like. Write a program that creates a list containing these items and then uses each function introduced in this chapter at least once.

In [ ]:

```python
Guests = ['Bhaskar', 'Spandan', 'Rajveer', 'Sichitaj', 'Abhisekh', 'Depaak']

print(Guests[0].upper())
print(Guests[0].lower())
print(Guests[0].title() +'Does nothing since it is already capital in first letter
print(Guests)
Guests.append('Abhirup')
print(Guests)
print(Guests.pop())
Guests.insert(3,'Abhirup')
print(Guests)
Guests.remove('Abhirup')
print(Guests)
Guests[0] = "Abhirup"
print(Guests)
del Guests[0]
print(Guests)

print(f'message to {Guests[3]}')

print(sorted(Guests))
print(sorted(Guests,reverse=True))
print(Guests)
Guests.reverse()
print(Guests)
Guests.sort()
print(Guests)
Guests.sort(reverse=0)
print(Guests)
print(len(Guests))
```

```
BHASKAR
bhaskar
BhaskarDoes nothing since it is already capital in first letter
['Bhaskar', 'Spandan', 'Rajveer', 'Sichitaj', 'Abhisekh', 'Depaak']
['Bhaskar', 'Spandan', 'Rajveer', 'Sichitaj', 'Abhisekh', 'Depaak', 'Abhirup']
Abhirup
['Bhaskar', 'Spandan', 'Rajveer', 'Abhirup', 'Sichitaj', 'Abhisekh', 'Depaak']
['Bhaskar', 'Spandan', 'Rajveer', 'Sichitaj', 'Abhisekh', 'Depaak']
['Abhirup', 'Spandan', 'Rajveer', 'Sichitaj', 'Abhisekh', 'Depaak']
['Spandan', 'Rajveer', 'Sichitaj', 'Abhisekh', 'Depaak']
message to Abhisekh
['Abhisekh', 'Depaak', 'Rajveer', 'Sichitaj', 'Spandan']
['Spandan', 'Sichitaj', 'Rajveer', 'Depaak', 'Abhisekh']
['Spandan', 'Rajveer', 'Sichitaj', 'Abhisekh', 'Depaak']
['Depaak', 'Abhisekh', 'Sichitaj', 'Rajveer', 'Spandan']
['Abhisekh', 'Depaak', 'Rajveer', 'Sichitaj', 'Spandan']
['Abhisekh', 'Depaak', 'Rajveer', 'Sichitaj', 'Spandan']
5
```

3-11. Intentional Error: If you haven’t received an index error in one of your programs yet, try to make one happen. Change an index in one of your programs to produce an index error. Make sure you correct the error before closing the program.

In [ ]:

```
Guests = ['Bhaskar', 'Spandan', 'Rajveer', 'Sichitaj', 'Abhisekh', 'Depaak']

print(Guests[100])
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[11], line 3
      1 Guests = ['Bhaskar', 'Spandan', 'Rajveer', 'Sichitaj', 'Abhisekh', 'Depaa
k']
----> 3 print(Guests[100])

IndexError: list index out of range
```