

## Functions

---

**Q1. Functions with pass by value. What will be the output?**

```
def CovidTimeline( x ) :  
    if (x < 2020):  
        print("Welcome to Before Covid life")  
    else:  
        print("Welcome to After Covid life")  
CovidTimeline(2019)  
CovidTimeline(2022)
```

Sol.

```
Welcome to Before Covid life  
Welcome to After Covid life
```

**Q2. Functions with pass by reference. Guess the output.**

```
def myFun(x) :  
    x[0] = 20  
list = [10, 11, 12, 13, 14, 15]  
myFun(list);  
print(list)
```

Sol.

```
[20, 11, 12, 13, 14, 15]
```

**Scope of variables. Local vs global.**

**Q3.**

**A. Predict the output**

```
def myfunc() :  
    x = 30  
    print(x)  
myfunc()
```

Sol.  
30

**B. Predict the output**

```
def myfunc():  
    x = 30  
    def myinnerfunc():  
        print(x)  
    myinnerfunc()  
myfunc()
```

Sol.  
30

**C. Predict the output**

```
x = 250  
def myfunc():  
    print(x)  
myfunc()  
print(x)
```

Sol.  
250  
250

**D. Predict the output**

```
x = 300  
def myfunc():  
    x = 200  
    print(x)  
myfunc()  
print(x)
```

Sol.  
200  
300

**E. Predict the output**

```
def myfunc():  
    global x  
    x = 100  
myfunc()  
print(x)
```

## Functions

---

Sol.  
100

### F. Predict the output

```
x = 400
def myfunc():
    global x
    x = 50
myfunc()
print(x)
```

Sol.  
50

### Functions with arguments.

#### Q4.

##### A) Predict the output

```
def my_function(fname, lname):
    print(fname + " " + lname)

my_function("Sachin", "Tendulkar")
```

Sol.  
Sachin Tendulkar

##### B) Predict the output

```
def my_function(fname, lname):
    print(fname + " " + lname)

my_function("Sachin")
```

Sol.  
**TypeError:** my\_function() missing 1 required positional argument: 'lname'

**Q5.**

**A) Functions with arbitrary arguments. Predict the output**

```
def my_function(*avenger):
    print("My fav avenger is " + avenger[2])
my_function("Cap", "Natasha", "Tony")
```

Sol.  
My fav avenger is Tony

**B) Functions with Keyword arguments. Predict the output**

```
def my_function(child3, child2, child1):
    print("The youngest child is " + child3)
my_function(child1 = "Ram", child2 = "Akbar", child3 =
"Tony")
```

Sol.  
The youngest child is Tony

**Q6. Predict the output**

**A) Function with arbitrary keyword args**

```
def greet(*names):
    for name in names:
        print("Hello", name)
greet("Monica", "Luke", "Steve", "John")
```

Sol.  
Hello Monica  
Hello Luke  
Hello Steve  
Hello John

**B) Function with default argument**

```
def my_function(country = "Norway"):
    print("I am from " + country)

my_function("Sweden")
my_function("India")
my_function() # no arg given my_function("Brazil")
```

Sol.

## Functions

---

```
I am from Sweden
I am from India
I am from Norway
```

### Q7. Predict the output

#### A) Passing list as arguments

```
def my_function(food):
    for x in food:
        print(x)
fruits = ["apple", "banana", "cherry"]
my_function(fruits)
```

Sol.

```
apple
banana
cherry
```

#### B) Return values

```
def my_function(x):
    return 5 * x
print(my_function(3))
print(my_function(5))
print(my_function(9))
```

Sol.

```
15
25
45
```

### Q8. Write a Python function that takes a list and returns a new list with unique elements of the

**Sample List :** [1,2,3,3,3,3,4,5]

**Unique List :** [1, 2, 3, 4, 5]

**Sol.**

```
def unique_list(l):
    x = []
    for a in l:
        if a not in x:
            x.append(a)
    return x
```

## Functions

---

```
print(unique_list([1,2,3,3,3,3,4,5]))
```

**Q9. Rohit is new to the English course. He needs to learn all the practice sheets done in the course. He asked her friend Anjali to provide a list of all the practice sheets and their associated marks. He needs to find average marks assigned to a problem. Question with maximum marks and lowest marks. Help Rohit by providing the python script for the same. Create a function to perform the above task**

**Input:**

3

1 Preposition problem 2.5

2 Article Problem 3

3 Spelling Problems 5

**Output:**

average marks per practice sheet is 3.5 marks

Spelling Problems is problem with maximum marks i.e. 5.0 marks

Preposition problem is problem with minimum marks i.e. 2.5 marks

**Description**

**Input:**

first line represents number of practice sheets 'k'

in next 'k' lines, first value represents practice number, followed by title of problem

and last value represents marks assigned for that problem.

**Output:**

average marks

problem title of highest mark problem

problem title of lowest mark problem

Sol.

```
def solution(N):
    assign_ID=[]
    course_title=[]
    marks=[]
    for i in range(N):
        S=list(input().split())
        last=len(S)-1
        assign_ID.append(int(S[0]))
        course_title.append(S[1:last])
```

## Functions

---

```
marks.append(float(S[last]))
avg_marks=sum(marks)/len(marks)
id_max=marks.index(max(marks))
id_min=marks.index(min(marks))
X=' '.join(course_title[id_max])
Y=' '.join(course_title[id_min])

print('average marks per assignment is {} marks'.format(avg_marks))
print('{} is problem with maximum marks i.e. {}
marks'.format(X,max(marks)))
print('{} is problem with minimum marks i.e. {}
marks'.format(Y,min(marks)))
```

### **Q10. Program a formula**

$$y=6x^2+3x+2, \text{ for } x=2$$

**Make a Python function that takes x as argument and returns y. Call the function for x=2 and print the answer.**

Sol.

```
def f(x):
    return 6*x**2 + 3*x + 2

y = f(2)
print (y)
```