

# MD RAJ ASSIGNMENT 4 IN PYTHON

**1. Write a function `convert_temperature(temp)` that converts temperature in Fahrenheit to degree Celsius**

```
def convert_temp(temp):  
    fahrenheit=(temp*9/5)+32  
    return fahrenheit  
  
temp=int(input("Enter Temperature in celsius"))  
  
print(convert_temp(temp))
```

**2. Write a function `number_sum(num)` that calculates the sum of the digits of a number**

```
def number_sum(num):  
    ans=0  
    while(num>0):  
        ans+=num%10  
        num//=10  
    return ans  
  
num=int(input("Enter a number"))  
  
ans = number_sum(num)  
  
print(f'the sum of the digits is {ans}')
```

# MD RAJ ASSIGNMENT 4 IN PYTHON

**3. Create a function factorial(n) that returns the factorial of a given number.**

```
def fact(num):  
    ans=1  
    while(num>0):  
        ans*=num  
        num-=1  
    return ans  
  
num=int(input("Enter a number"))  
ans=fact(num)  
  
print(f"The factorial of {num} is {ans}")
```

**4. Write a function prime\_number(n) that displays whether a number entered by the user is prime or not.**

```
def isPrime(num):  
    if num<=1:  
        return False  
    for i in range(2,int(num**0.5)+1):  
        if num%i==0:  
            return False  
    return True  
  
num=int(input("Enter a number"))  
  
if isPrime(num):  
    print("Its a prime number")  
else:  
    print("Its not a prime number")
```

# MD RAJ ASSIGNMENT 4 IN PYTHON

## 5. Use a lambda function to: • Square a number • Add two numbers

```
num=int(input("Enter a number"))
square=lambda num: num**2
print(f'The square of {num} is {square(num)}')
```

```
num1=int(input("Enter first number"))
num2=int(input("Enter second number"))
```

```
add=lambda x,y:num1+num2
print(f'The sum of {num1}+{num2} is {add(num1,num2)}')
```

#x and y are parameters and num1+num2 is the operation

#add is the function variable which will store the anonymous function defined using lambda

## 6. Create a list of integers. Perform the following: • Append elements • Remove an element • Sort the list in descending order • Find the sum and average of the list

```
numbers=[1,2,3,4,5,6,7]#this is a list of numbers
```

```
print(numbers)#[1, 2, 3, 4, 5, 6, 7]
```

```
numbers.append(8)
```

```
print(numbers)#[1, 2, 3, 4, 5, 6, 7, 8] append means adding an item in the last index
```

```
numbers.remove(4)
```

```
print(numbers)#[1, 2, 3, 5, 6, 7, 8] 4 is removed from the list
```

```
numbers.sort(reverse=True)
```

# MD RAJ ASSIGNMENT 4 IN PYTHON

`print(numbers)`#it will first sort in ascending order then it will reverse the list to make it in descending order

`#[8, 7, 6, 5, 3, 2, 1]`

`numbers.extend([4,9,10])`#extend takes only one argument but i have to send multiple numbers so i made it a list then sending now python will consider it one argument as it is a single list

`numbers.sort()`#[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

`print(numbers)`

`#creating a list from 1 to 10 numbers`

`ans=0`

`for i in numbers:`

`ans+=i` #the sum from 1 to 10 will be stored in ans

`avg=ans/len(numbers)`#then avg will be calculated

`print(f'the average is {avg}')`

## **7.Create a tuple of 5 items. Demonstrate slicing, indexing, and unpacking of the tuple.**

`myTuple=(1,2,3,4,5)`

`print(myTuple[2])`#3, we can access tuples with index numbers it starts with index0 and we can also do negative indexing it starts with the last index say -1

`print(myTuple[-2])`#4

`print(myTuple[2:4])` #(3, 4)#it returns a portion of the tuple from which the slicing starts and to which slicing should be done here slicing starts with index 2 and till index 4

`myInfo=("raj",25,"hacker")`

# MD RAJ ASSIGNMENT 4 IN PYTHON

```
name,age,profession=myInfo
```

```
print(name)#raj
```

```
print(age)#25
```

```
print(profession)#hacker
```

```
#i unpacked the tuple myInfo and assigned it to 3 different corresponding variables
```

**8.Create a set of fruits. Add a new fruit, remove one fruit, and demonstrate that duplicates are automatically removed.**

```
fruitsSet={"apple","banana","mango","litchi","apple"}
```

```
print(fruitsSet)
```

```
#{'litchi', 'mango', 'apple', 'banana'}
```

```
#here we can see that only unique values are printed and also it is unordered.
```

```
fruitsSet.add("chiku")
```

```
print(fruitsSet)
```

```
#{'chiku', 'litchi', 'apple', 'banana', 'mango'}
```

```
#using add function we can add only a single item at a time
```

```
#to add multiple items we use update method
```

```
fruitsSet.update(["orange","strawberry","etc"])
```

```
print(fruitsSet)
```

```
#{'banana', 'etc', 'chiku', 'mango', 'orange', 'apple', 'strawberry', 'litchi'}
```

```
fruitsSet.remove("apple")
```

```
print(fruitsSet)#{'banana', 'orange', 'etc', 'chiku', 'strawberry', 'litchi', 'mango'} apple is removed from the fruitsSet
```

# MD RAJ ASSIGNMENT 4 IN PYTHON

**9.Create a dictionary of student names and their marks. Add a new entry, update an existing one, delete an entry, and find the student with the highest marks.**

```
students={
    "raj":98,
    "ramesh":89,
    "himesh":23,
    "roshni":54,
    "saran":89
}
print(students)
#{'raj': 98, 'ramesh': 89, 'himesh': 23, 'roshni': 54, 'saran': 89}

#created a dictionary of students where names are the keys and marks are their corresponding values

#adding a student in the dictionary

students["neha"]=80

print(students)
#{'raj': 98, 'ramesh': 89, 'himesh': 23, 'roshni': 54, 'saran': 89, 'neha': 80}

students["raj"]=100

print(students)
#{'raj': 100, 'ramesh': 89, 'himesh': 23, 'roshni': 54, 'saran': 89, 'neha': 80}

#updated the marks of raj

del students["ramesh"]
```

# MD RAJ ASSIGNMENT 4 IN PYTHON

```
print(students)
```

```
#{'raj': 100, 'himesh': 23, 'roshni': 54, 'saran': 89, 'neha': 80} ramesh has deleted from the dictionary
```

```
highest_student=max(students,key=students.get)
```

```
highestMarks=students[highest_student]
```

```
print(f"Topper is {highest_student} with {highestMarks} marks")
```

```
#Topper is raj with 100 marks
```

**10. Write a function that prints the reverse of a number (like if a user enters 4587 it gives 7854) .**

```
num=int(input("Enter a number")) #1234
```

```
rev=0
```

```
while(num>0):#1234 is greater than 0 true ,123>0 ,12>0 2>0 ,0>0->false ,out from loop
```

```
digit=num%10 #1234%10=4 ,123%10=3 ,12%10=2 1%10=1
```

```
rev=rev*10+digit #0*10+4=4 ,4*10+3=43 ,43*10+2=432 432*10+1=4321
```

```
num=num//10 #1234//10=123 ,123//10=12 ,12//10=1 ,2//10=0
```

```
print(rev)#printing rev value
```