

Python Practical’s

# TASK 5

Smit Joshi | 19-09-2023

View On [**github.com/smit-joshi814**](https://github.com/smit-joshi814/Learning-python/tree/main/collage/Task5)

# Practical 1

1. Write a python program for performing following operations and display results after each

operation:

(i) Create dictionary with details of 5 students : Rollno, name, semester, course, percentage.

(ii) Update semester of student3 to 2.

(iii) Delete the data of 4th student.

(iv) Copy this dictionary to dict2

(v) Display the length of both the dictionaries

dictionary={

    "student1":{

        "RollNo":1,"name":"smit","semester":7,"course":"mscit","percentage":90

    },

    "student2":{

        "RollNo":2,"name":"tejasv","semester":6,”course":"bscit","percentage":80

    },

    "student3":{

        "RollNo":3,"name":"switi","semester":8,"course":"mscit","percentage":89

    },

    "student4":{

       "RollNo":4,"name":"tabbssum","semester":5,"course":"bscit","percentage":50

    },

    "student5":{

        "RollNo":5,"name":"nisha","semester":7,"course":"mscit","percentage":40

    },

}

dictionary["student3"]["semester"]=2

print()

print(dictionary["student3"])

print()

del dictionary["student4"]

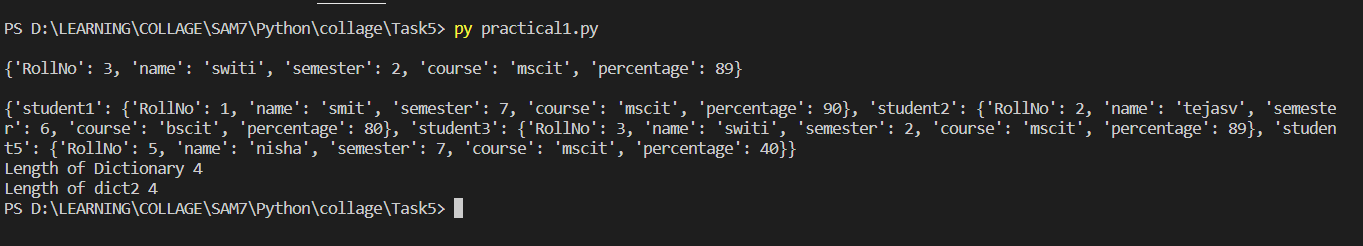
dict2=dictionary.copy()

print(dict2)

print("Length of Dictionary",len(dictionary))

print("Length of dict2",len(dict2))

### Output:



# Practical 2

Create a dictionary with key as country and their capitals as values with 5 countries spain,

france, germany, norway. Perform the following operations on this dictionary.

(i) Print the capital of germany

(ii) Remove norway and its capital from dictionary

(iii) Take one country from user and print the capital if it is present else print it is not present

in the dictionary.

(iv) Add India with capital in the dictionary

country={

    "capitals":{

        "spain":"Madrid",

        "france":"Paris",

        "germany":"Berlin",

        "norway":"Oslo"

    }

}

print(country["capitals"]["germany"])

del country["capitals"]["norway"]

print(country)

u\_in=input("Enter Country ")

for i in country["capitals"]:

    if u\_in==i:

        print(f"{i}:{country['capitals'][i]}")

        break

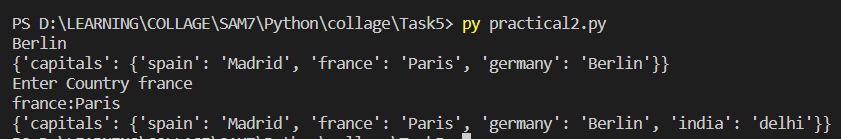
else:

    print("it is not present in the dictionary")

country["capitals"]["india"]="delhi"

print(country)

### Output:



# Practical 3

Write a menu driven program to implement the stack with operations push,pop and display.

LIMIT=50

stack=[]

top=-1

def push(value):

    global top

    global LIMIT

    if top < LIMIT:

        top+=1

        stack.append(value)

    else:

        print("stack is full")

def pop():

    global top

    if top!=-1:

        val=stack[top]

        top-=1

        return val

    else:

        return None

def display():

    global top

    if top!=-1:

        for i in range(top+1):

            print(stack[i],end=" ")

    else:

        print("Stack is Empty")

while True:

    print()

print("Enter 1 For push()",end="\t")

    print("Enter 2 For pop()",end="\t")

    print("Enter 3 To Display()",end="\t")

    print("Enter 4 To Exit()")

    choice=input("Enter Choice: ")

    if choice=="1":

        val=input("Enter value: ")

        push(val)

    elif choice=="2":

        print(f"{pop()} deleted")

    elif choice=="3":

        display()

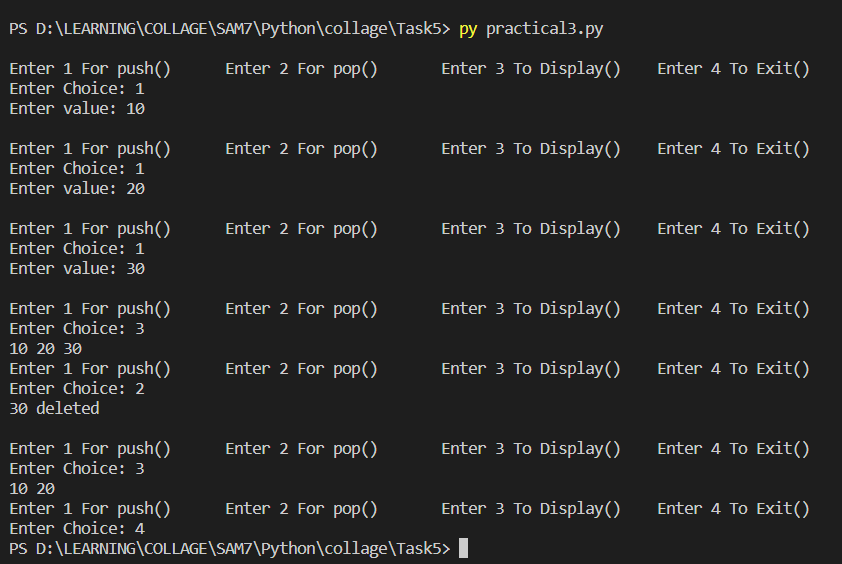
    elif choice=="4":

        break

    else:

        print("invalid choice")

### Output:



# Practical 4

Write a menu driven program to implement the queue with operations insertion , deletion And display.

queue=[]

LIMIT=10

FIRST=-1

LAST=-1

def insert(value):

    global LIMIT,FIRST,LAST

    if FIRST < LIMIT:

        if FIRST==-1 and LAST==-1:

            FIRST=0

            LAST=0

        else:

            LAST+=1

        queue.append(value)

    else:

        print("Queue is Full")

def deletion():

    global FIRST,LAST

    if FIRST==-1:

        print("Queue is Empty")

        return

    else:

        val=queue[FIRST]

        if FIRST==LAST:

            FIRST=-1

            LAST=-1

        else:

            FIRST+=1

        return val

def display():

    global FIRST,LAST

    if FIRST!=-1:

        for i in range(FIRST,LAST+1):

            print(queue[i],end=" ")

    else:

        print("Quue is Empty")

while True:

    print()

    print("Enter 1 to Insert()",end="\t")

    print("Enter 2 to Delete()",end="\t")

    print("Enter 3 to Display()",end="\t")

    print("Enter 4 to Exit()")

    choice=input("Enter choice: ")

    if choice=="1":

        val=input("Enter Value: ")

        insert(val)

    elif choice=="2":

        print(deletion(),"Deleted")

    elif choice=="3":

        display()

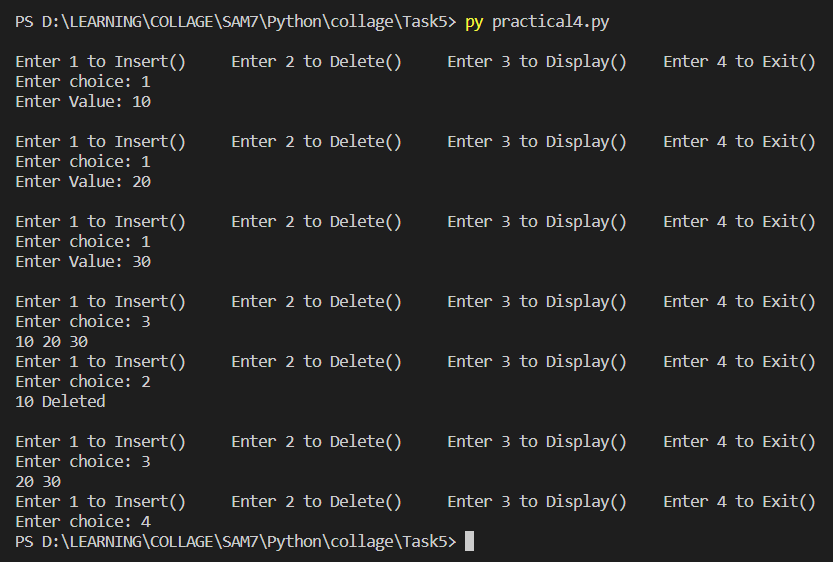
    elif choice=="4":

        break

    else:

        print("invalid choice")

### Output:



# Practical 5

Write a python program to demonstrate the concept of default arguments. Take the

employee data from user: name, department and basic\_salary. Write a function to to

calculate total salary of the employee where total salary is sum of basic salary, DA and

HRA. DA is 10% of basic salary and HRA is 15% of basic salary. If basic salary is missing,

take default argument of basic salary as Rs. 9000

def cacTotalSalary(employeeName,department,basic=9000):

    da=basic\*0.10

    hra=basic\*0.15

    totalSalary=basic+da+hra

    print(f"Name: {name} \nDepartment: {department}\n Total Salary: {totalSalary}")

name=input("Enter Name ")

department=input("Enter Department ")

basic=int(input("Enter Basic "))

cacTotalSalary(name,department,basic)

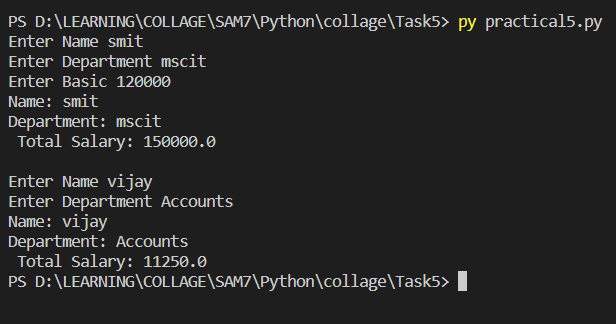
print()

name=input("Enter Name ")

department=input("Enter Department ")

cacTotalSalary(name,department)

### Output:



# Practical 6

Take two values of feet and inches from user. Create a function to add the values of feet

with feet and inch with inch. Display the valid total result. (Inch should not be more than 11)

def sumFitInch(feet1,feet2,inch1,inch2):

    Sumfeet=feet1+feet2

    SumInchs=inch1+inch2

    if SumInchs>12:

        Sumfeet+=SumInchs//12

        SumInchs=SumInchs%12

    return Sumfeet,SumInchs

feet1=int(input("Enter value of Feet 1: "))

feet2=int(input("Enter value of Feet 2: "))

inch1=int(input("Enter value of Inch 1: "))

inch2=int(input("Enter value of Inch 2: "))

if inch1>11 or inch2 >11:

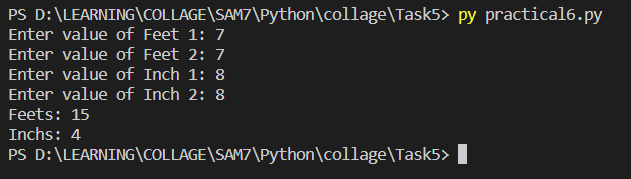
    print("Inch should not be more than 11")

else:

    totalFeets,totalInchs=sumFitInch(feet1=feet1,feet2=feet2,inch1=inch1,inch2=inch2)

    print(f"Feets: {totalFeets} \nInchs: {totalInchs}")

### Output:



# Practical 7

Write a function to find whether the number is armstrong number or not

def isArmstrong(number):

    sum=0

    temp=number

    while temp>0:

        digit=temp%10

        sum+=digit\*\*3

        temp//=10

    if sum==number:

        return True

    else:

        print(sum)

        return False

number=int(input("Enter number: "))

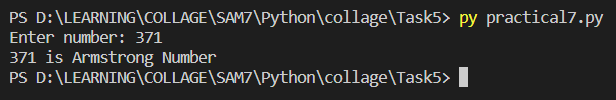
if isArmstrong(number=number):

    print(f"{number} is Armstrong Number")

else:

    print("Number is Not Armstrong")

### Output:



# Practical 8

Write a function to add the values of two list and store that in third list and display the result.

def listOperation(list1,list2):

    if len(list1)!=len(list2):

        print("List Should be of the same size ")

        return

    else:

        list3=[]

        for i in range(len(list1)):

            list3.append(list1[i]+list2[i])

        return list3

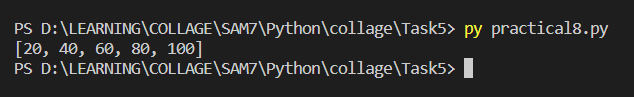
list1=[10,20,30,40,50]

list2=[10,20,30,40,50]

list3=listOperation(list1,list2)

print(list3)

### Output:



# Practical 9

Write a recursive function taking one integer argument, if argument is zero return zero else

return sum of all the predecessor of that argument

def sumOfPredecesor(number):

    if number == 0:

        return 0

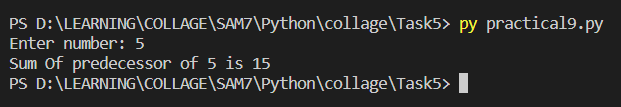
    else:

        return number+ sumOfPredecesor(number-1)

number=int(input("Enter number: "))

print(f"Sum Of predecessor of {number} is {sumOfPredecesor(number)}")

### Output:



# Practical 10

Write a recursive function for printing the fibonacci series. The argument n will be passed

to the function, where n is number of terms in the series

def fibonacci(n):

  if n <= 1:

    return n

  else:

    return (fibonacci(n - 1)+fibonacci(n-2))

n = int(input("Enter N: "))

for i in range(n+1):

    print(fibonacci(i))

### Output:

