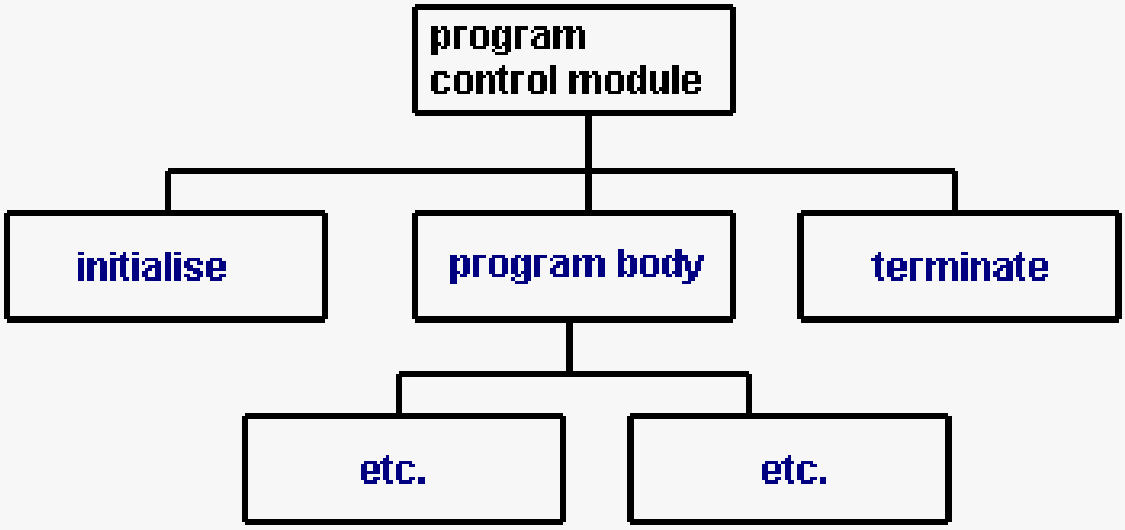
PROGRAMMING IN PYTHON

Lab Exercise - 8

“Dictionary DS & FUNCTIONS”



Name – Purushottam Kumar

ID – 2041

Date - 27th Feb 2021

❖ DICTIONARY BASED PROBLEMS

1) Write a program that uses a dictionary to store states and their codes. Print the dictionary.
   (First create an empty dictionary and add keys and values by getting user input)

```
D={}
N=int(input("TOTAL NO. OF STATES : "))
for i in range(N):
    st_code=input("States Code : ")
    st_name=input("States name : ")
    if st_code not in D:
        D[st_code]=st_name
print(D)
```

OUTPUT :

```
TOTAL NO. OF STATES : 4
States Code : BR
States name : Bihar
States Code : JH
States name : Jharkhand
States Code : TN
States name : Tamilnadu
States Code : UP
States name : Uttar Pradesh
{'TN': 'Tamilnadu', 'BR': 'Bihar', 'UP': 'Uttar Pradesh', 'JH': 'Jharkhand'}


...Program finished with exit code 0
Press ENTER to exit console.
```

2) WAP to search for the presence or absence of a state code from the dictionary created in program 1.

```
#Search in The States

Code=input("Type State Code to search : ")
if Code in D:
    print("State Found ",Code ," : ", D[Code])
else:
    print("State Code Not Found")
```

OUTPUT :

```
Type State Code to search : BR
State Found  BR  :  Bihar
```

3) Write a program to create a dictionary using dictionary comprehension. The keys must be odd numbers in the range 1-100 and the values must be the cube of the key

```
Num={x:x*x*x for x in range(1,101) if x%2==1}
for i in sorted(Num.keys()):
    print(i,Num[i])
```

OUTPUT :

```
$python main.py
(1, 1)
(3, 27)
(5, 125)
(7, 343)
(9, 729)
(11, 1331)
(13, 2197)
(15, 3375)
(17, 4913)
(19, 6859)
(21, 9261)
(23, 12167)
(25, 15625)
(27, 19683)
(29, 24389)
(31, 29791)
(33, 35937)
(35, 42875)
(37, 50653)
(39, 59319)
(41, 68921)
(43, 79507)
(45, 91125)
(47, 103823)
(49, 117649)
(51, 132651)
(53, 148877)
(55, 166375)
(57, 185193)
(59, 205379)
(61, 226981)
(63, 250047)
(65, 274625)
(67, 300763)
(69, 328509)
(71, 357911)
(73, 389017)
(75, 421875)
(77, 456533)
(79, 493039)
(81, 531441)
(83, 571787)
(85, 614125)
(87, 658503)
(89, 704969)
(91, 753571)
(93, 804357)
(95, 857375)
(97, 912673)
(99, 970299)
```

4) Write a program that calculates fib(n) using dictionary.

```
# Without Recursion
D={1:0, 2:1}
def fibo(n):
    a = 0
    b = 1
    if n == 0:
        return 0
    elif n == 1:
        return b
    else:
        for i in range(3, n+3):
            c = a + b
            a = b
            b = c
            D[i]=b
N=int(input("Enter Total Terms : "))
fibo(N)
print(D[N])
```

```
# Using Recursion

D={0:0, 1:1}
def fibo(n):
    if n not in D:
        val=fibo(n-1)+fibo(n-2)
        D[n]=val
    return D[n]
N=int(input("Enter Term : "))
fibo(N)
print(D)
print(N,"th Term : ",D[N-1])
```

OUTPUT :

```
Shell                                                    Clear
Enter Term : 10
{0: 0, 1: 1, 2: 1, 3: 2, 4: 3, 5: 5, 6: 8, 7: 13, 8: 21, 9: 34, 10: 55}
10 th Term :  34
>
```

5) Write a program to store a matrix using dictionary. The key is (row, column) information and values is the number in that matrix location.

```
row=int(input("Total Rows : "))
column=int(input("Total Columns : "))
matrix = {(i, j): input("ENTER DATA : ") for i in range(row) for j in range(column)}
for i in sorted(matrix.keys()):
    print(i,":",matrix[i])
```

OUTPUT :

```
Total Rows : 2
Total Columns : 3
ENTER DATA : 1
ENTER DATA : 29
ENTER DATA : 6
ENTER DATA : 7
ENTER DATA : 8
ENTER DATA : 9
(0, 0) : 1
(0, 1) : 29
(0, 2) : 6
(1, 0) : 7
(1, 1) : 8
(1, 2) : 9


...Program finished with exit code 0
Press ENTER to exit console.
```

PART-2 “FUNCTIONS BASED PROBLEM”

1) Write a program using functions to check whether two numbers are equal or not.

main.py

Run

```
def checkequal():
    A=eval(input("Enter First Number : "))
    B=eval(input("Enter Second Number : "))
    if (A==B):
        print("Numbers are equal")
    else:
        print("Numbers are not equal")
checkequal()

```

OUTPUT :

Shell

```
Enter First Number : 15
Enter Second Number : 39
Numbers are not equal

Enter First Number : 24
Enter Second Number : 24.0
Numbers are equal

Enter First Number : 36
Enter Second Number : 36
Numbers are equal
```

2) Write a program to swap two numbers.

main.py

Run

```
def swap(a,b):
    a=a*b
    b=a/b
    a=a/b
    return (a,b)
A=eval(input("Enter First Number : "))
B=eval(input("Enter Second Number : "))
print("Before Swapping A=",A," B=",B)
A,B=swap(A,B)
print("After Swapping A=",A," B=",B)
```

OUTPUT :

```
Enter First Number : 26
Enter Second Number : 37
Before Swapping A= 26  B= 37
After Swapping A= 37  B= 26
```

3) Write a program to return the complete address of a person. (Accept separate information about door no, building name, street name, locality name, city name, state name, pin code)

```
def ReturnAddress(Door,Building,Street,Local,Ct,St,Pin):
    return (str(Door)+','+Building+',\n'+Street+','+Local+' '+Ct+',\n'+St+'-'+str(Pin))
Door_no =int(input("Enter Door No : "))
Building_Nm =input("Enter Building Name : ")
Street_nm =input("Enter Street Name : ")
Locality=input("Enter Locality : ")
City_nm=input("Enter City Name : ")
State_nm=input("Enter State Name : ")
Pin_cd=int(input("Enter Pin Code : "))

Full_Address=ReturnAddress(Door_no,Building_Nm,Street_nm,Locality,City_nm,State_nm,Pin_cd)
print("\nFULL ADDRESS : \n",Full_Address)
```

OUTPUT :

```
Enter Door No : 36
Enter Building Name : Nirmla Sadan
Enter Street Name : LBT College Road
Enter Locality : Chini Mill
Enter City Name : Buxar
Enter State Name : Bihar
Enter Pin Code : 802103

FULL ADDRESS :
 36,Nirmla Sadan,
LBT College Road,Chini Mill Buxar,
Bihar-802103
```

4) Write a program to return the average of its arguments.

```
def FindAvg(args):
    sum1=0
    for i in args:
        sum1+=i
    avg=sum1/len(args)
    return avg
Num=[int(x) for x in input("Enter Numbers to find averarge : ").split()]
print(type(Num))
Average=FindAvg(Num)
print("Average = ",Average)
```

Shell Clear

```
Enter Numbers to find averarge : 2 4 63 7
Average =  19.0
Enter Numbers to find averarge : 1 2 6 4 5
Average =  3.6
```

5) Write a program using function and return statement to check whether a number is Armstrong number or not.

main.py

```
def CheckArmstrong(A):
    temp=A
    sum1=0
    while(A>=1):
        rem=A%10
        sum1=sum1+(rem*rem*rem)
        A//=10
    if(sum1==temp):
        print("Yes ! It is Armstrong")
    else:
        print("No! It is Not Armstrong")
N=int(input("Enter Number to check Armstrong : "))
CheckArmstrong(N)
```

OUTPUT

Shell

```
Enter Number to check Armstrong : 153
Yes ! It is Armstrong
Enter Number to check Armstrong : 239
No! It is Not Armstrong
Enter Number to check Armstrong : 370
Yes ! It is Armstrong
```

6) Write a program to reverse a string using recursion.

main.py

```
def reverse_string(str1):
    if len(str1)==0:
        return str1
    else:
        return str1[-1] + reverse_string(str1[0:-1])
N=input("Enter A String : ")
print("Reverse of ",N," is : ",reverse_string(N))

```

OUTPUT

```
Enter A String : MANOHAR
Reverse of  MANOHAR  is :  RAHONAM
>
```

7) Write a program to find the largest of three numbers.

main.py

Run

```
def Find_Max(a,b,c):
    if (a>b and a>c):
        return a
    elif (b>c):
        return b
    else:
        return c
A,B,C=int(input("First ")),int(input("Second ")),int(input("Third "))
print("Biggest Number is : ",Find_Max(A,B,C))

```

OUTPUT

Shell

Clear

```
First 36
Second 2
Third 92
Biggest Number is :  92
>
```

----------*----------*----------