

**Practical No: 4**

1. To check whether string is palindrome or not using function recursion

Code:

```
def isPalindrome(string):  
    if len(string)<=1:  
        return True  
    else:  
        return string[0]==string[-1] and  
isPalindrome(string[1:-1])  
  
while True:  
    print("\nPress 9 to quit.\n")  
    s = input("Enter a string to check if it's  
Palindrome or not\n");  
    s = s.lower();  
    if s=="9":  
        break  
    elif(isPalindrome(s) == True):  
        print("String is Palindrome.\n");  
    else:  
        print("String is not Palindrome.\n");  
  
print("Thank you!!!")
```

Output:

```
C:\Users\mca_dept\Desktop\Python Pracs\Lab4>python program1.py
Press 9 to quit.
Enter a string to check if it's Palindrome or not
Hammah
String is Palindrome.

Press 9 to quit.
Enter a string to check if it's Palindrome or not
Hammad
String is not Palindrome.

Press 9 to quit.
Enter a string to check if it's Palindrome or not
AnnA
String is Palindrome.

Press 9 to quit.
Enter a string to check if it's Palindrome or not
aNna
String is Palindrome.

Press 9 to quit.
Enter a string to check if it's Palindrome or not
Nishita
String is not Palindrome.

Press 9 to quit.
Enter a string to check if it's Palindrome or not
9
Thank you!!!
C:\Users\mca_dept\Desktop\Python Pracs\Lab4>_
```

## 2. To find Fibonacci series using recursion

Code:

```
fibonacci_cache = {}

def fibonacci(n):
    if n in fibonacci_cache:
        return fibonacci_cache[n]
    if n==1:
        value = 1
```

```
        elif n==2:
            value = 1
        elif n>2:
            value = fibonacci(n-2)+fibonacci(n-1)
        fibonacci_cache[n] = value
        return value

while True:

    print("\nPress 0 to quit.\n")

    n = int(input("Enter range for fibonacci
series:\n"))

    if n==0:
        break
    else:
        for i in range(1,n+1):
            print("\t fibonacci of ",i," =
",fibonacci(i))

print("Thank you!!!")
```

**Output:**

```

fibonacci of 46 = 1836311903
fibonacci of 47 = 2971215073
fibonacci of 48 = 4807526976
fibonacci of 49 = 7778742049
fibonacci of 50 = 12586269025
fibonacci of 51 = 20365011074
fibonacci of 52 = 32951280099
fibonacci of 53 = 53316291173
fibonacci of 54 = 86267571272
fibonacci of 55 = 139583862445
fibonacci of 56 = 225851433717
fibonacci of 57 = 365435296162
fibonacci of 58 = 591286729879
fibonacci of 59 = 956722026041
fibonacci of 60 = 1548008755920
fibonacci of 61 = 2504730781961
fibonacci of 62 = 4052739537881
fibonacci of 63 = 6557470319842
fibonacci of 64 = 10610209857723
fibonacci of 65 = 17167680177565
fibonacci of 66 = 27777890035288
fibonacci of 67 = 44945570212853
fibonacci of 68 = 72723460248141
fibonacci of 69 = 117669030460994
fibonacci of 70 = 190392490709135
fibonacci of 71 = 308061521170129
fibonacci of 72 = 498454011879264
fibonacci of 73 = 806515533049393
fibonacci of 74 = 1304969544928657
fibonacci of 75 = 2111485077978050
fibonacci of 76 = 3416454622906707
fibonacci of 77 = 5527939700884757
fibonacci of 78 = 8944394323791464
fibonacci of 79 = 14472334024676221
fibonacci of 80 = 23416728348467685
fibonacci of 81 = 37889062373143906
fibonacci of 82 = 61305790721611591
fibonacci of 83 = 99194853094755497
fibonacci of 84 = 160500643816367088
fibonacci of 85 = 259695496911122585
fibonacci of 86 = 420196140727489673
fibonacci of 87 = 679891637638612258
fibonacci of 88 = 1100087778366101931
fibonacci of 89 = 1779979416004714189
fibonacci of 90 = 2880067194370816120
fibonacci of 91 = 4660046610375530309
fibonacci of 92 = 7540113804746346429
fibonacci of 93 = 12200160415121876738
fibonacci of 94 = 19740274219868223167
fibonacci of 95 = 31940434634990099905
fibonacci of 96 = 51680708854858323072
fibonacci of 97 = 83621143489848422977
fibonacci of 98 = 135301852344706746049
fibonacci of 99 = 218922995834555169026
fibonacci of 100 = 354224848179261915075

```

### 3. To find binary equivalent of number using recursion

Code:

```
Binary_Storage = []
```

```
count = 0
```

```
def convertB(n,count):
```

```
        if(n>0):
            count=count+1
            Binary_Storage.append(n%2)
            convertB(n//2,count)

while True:

    print("\nPress 0 to quit.\n")

    s = int(input("Enter a decimal number to convert it
into binary number:\n"))

    if s==0:
        break

    else:
        convertB(s,-1)

        Binary_Storage.reverse()

        print(*Binary_Storage)

        Binary_Storage.clear()

print("Thank you!!!")
```

**Output:**

```
C:\Users\mca_dept\Desktop\Python Pracs\Lab4>python program3.py
Press 0 to quit.
Enter a decimal number to convert it into binary number:
4
1 0 0
Press 0 to quit.
Enter a decimal number to convert it into binary number:
215
1 1 0 1 0 1 1 1
Press 0 to quit.
Enter a decimal number to convert it into binary number:
256
1 0 0 0 0 0 0 0 0
Press 0 to quit.
Enter a decimal number to convert it into binary number:
255
1 1 1 1 1 1 1 1
Press 0 to quit.
Enter a decimal number to convert it into binary number:
765
1 0 1 1 1 1 1 1 0 1
Press 0 to quit.
Enter a decimal number to convert it into binary number:
123125345
1 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1 1 0 0 1 1 0 0 0 0 1
Press 0 to quit.
Enter a decimal number to convert it into binary number:
345
1 0 1 0 1 1 0 0 1
Press 0 to quit.
Enter a decimal number to convert it into binary number:
12
1 1 0 0
Press 0 to quit.
Enter a decimal number to convert it into binary number:
-
```

4. To use lambda function on list to generate filtered list, mapped list and reduced list

Code:

```
from functools import reduce
```

```
class AnonDemo:
```

```
    def lambdaDemo(self):
```

```
sum = lambda x, y : x + y

a=[2,3,5,6,7]

b=[6,9,2,1,8]

print(sum(a,b))

print("Sum of elements using lambda function
:- ",sum(3,5))

def mapDemo(self):

    items=[1,2,3,4,5]

    squared = list(map(lambda x : x**2, items))

    print("Squared Items using map function :-
",squared)

def filterDemo(self):

    number_list = range(-10, 10)

    less_than_zero = list(filter(lambda x: x < 0,
number_list))

    print("Less than zero elements using filter
function :- ",less_than_zero)

def reduceDemo(self):

    product = reduce((lambda x, y: x * y), [1, 2,
3, 4])

    print("Product of all items using reduce
function :- ",product)

ob = AnonDemo()

ob.lambdaDemo()

ob.mapDemo()

ob.filterDemo()

ob.reduceDemo()
```

Output:

```
PS C:\Users\hamma\Desktop\PythonPracs> python program4.py
[2, 3, 5, 6, 7, 6, 9, 2, 1, 8]
Sum of elements using lambda function :- 8
Squared Items using map function :- [1, 4, 9, 16, 25]
Less than zero elements using filter function :- [-10, -9, -8, -7, -6, -5, -4, -3, -2, -1]
Product of all items using reduce function :- 24
PS C:\Users\hamma\Desktop\PythonPracs>
```

5. Convert the temperature in Celsius to Fahrenheit in list using anonymous function

Code:

```
class Cf:

    def convert(self):

        n = int(input("Enter number of inputs :-
\n"))

        a = []

        for i in range(0,n):

            c = float(input("Enter celsius to
convert :- \n"))

            a.append(c)

        return a

ob = Cf()

t = ob.convert()

f = list(map(lambda x: (float(9)/5)*x + 32, t))

print(f)
```

Output:



```
PS C:\Users\hamma\Desktop\PythonPracs> python program5.py
Enter number of inputs :-
4
Enter celsius to convert :-
2
Enter celsius to convert :-
3
Enter celsius to convert :-
4
Enter celsius to convert :-
5
[35.6, 37.4, 39.2, 41.0]
PS C:\Users\hamma\Desktop\PythonPracs>
```

6. To create module in python and access functions of the module by importing it to another file/module. (Calculator program)

Code:

#name of the main file is calc.py

import calculator

x = float(input("Enter first number: "))

y = float(input("Enter second number: "))

while True:

print("Select one of the operations \n1.Addition \n2.Subtract  
\n3.Multiplication \n4.Division \n5.Mod \n6.Exit")

select = int(input(" "))

if(select == 1):

print(calculator.Add(x, y))

elif(select == 2):

```
print(calculator.Sub(x, y))
```

```
elif(select == 3):
```

```
print(calculator.Mul(x, y))
```

```
elif(select == 4):
```

```
print(calculator.Div(x, y))
```

```
elif(select == 5):
```

```
print(calculator.Mod(x, y))
```

```
elif(select == 6):
```

```
break
```

```
else:
```

```
print("Invalid input")
```

file to be imported: calculator.py

```
def Add(a, b):
```

```
    r = a + b
```

```
    return r
```

```
def Sub(a, b):
```

```
    r = a - b
```

```
    return r
```

```
def Mul(a, b):
```

```
    r = a * b
```

```
    return r
```

```
def Div(a, b):
```

```
    if (a != 0 and b != 0):
```

```
        r = a // b
```

```
        return r
```

```
    else:
```

```
        return ("Divion is not possible , its not defined ")
```

```
def Mod(a, b):
```

```
    r = a % b
```

```
    return r
```

Output:

```
Enter first number: 45.7
Enter second number: 98.7
Select one of the operations
1.Addition
2.Subtract
3.Multiplication
4.Division
5.Mod
6.Exit
1
144.4
Select one of the operations
1.Addition
2.Subtract
3.Multiplication
4.Division
5.Mod
6.Exit
2
-53.0
Select one of the operations
1.Addition
2.Subtract
3.Multiplication
4.Division
5.Mod
6.Exit
3
4510.59
- - - - -
Select one of the operations
1.Addition
2.Subtract
3.Multiplication
4.Division
5.Mod
6.Exit
4
0.0
Select one of the operations
1.Addition
2.Subtract
3.Multiplication
4.Division
5.Mod
6.Exit
5
45.7
```