Kalyani Government Engineering College

Department of Computer Application

Python Programming Lab – MCAN191
Year: 2023-2024
Semester: 1st Semester
Assignment 7
Name: Madhusudan Chand **Roll:** 13

7.1 Write a function find_max accepts three numbers as arguments and returns the largest number among three. Write another function main, in main() function accept three numbers from user and call find_max

```
Code:
def find_max(a,b,c):
    return max(a,b,c)
def main():
    a,b,c =[int(x) for x in input("Enter three number\n").split()]
    print("Max is ",find_max(a,b,c))
if __name__ == '__main__':
    main()

output:
Enter three number
28 -89 15
Max is  28
```

7.2 Write a function, is_vowel that returns the value true if a given character is a vowel, and otherwise returns false. Write another function main, in main() function accept a string from user and count number of vowels in that string.

```
Code:
def is_vowel(c):
    s="aeiuAEIOU"
    if c in s:
        return True
    return False

def main():
    x=input("Enter the string\n")
    d=[]
```

```
    for i in x:
        if is_vowel(i):
            d.append(i)
    print("Number of vowels",len(d))
if __name__ == "__main__":
    main()

output:
Enter the string
madhusudan
Number of vowels 4
```

7.3 Write a function named is_prime, which takes an integer as an argument and returns trueif the argument is a prime number, or false otherwise. Also, write the main function that displays prime numbers between 1 to 500.

```
Code:
def is_prime(c):
    flag=0
    for i in range(3,500,2):
        if c%i==0:
            break

    if i==c:
        return True
    else:
        return False

def main():
    d=[]
    d.append(2)
    for i in range(3,500,2):
        if is_prime(i):
            d.append(i)

    print(d)

if __name__ == "__main__":
    main()
```

output:

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499]

7.4 Write a function in python to find the sum of the cube of elements in a list. The list isreceived as an argument to the function, in turn, the function must return the sum. Write the main function which invokes the above function.

Code:

```
def sum_cube(l):
    return sum([x**3 for x in l])

def main():
    s=[int(x) for x in input("Enter the elements of the list\n").split()]
    print("sum of the cube of elements ",sum_cube(s))
    pass

if __name__ == "__main__":
    main()
```

ouput:

```
Enter the elements of the list
1 2 3
sum of the cube of elements  36
```

7.5 Write the definition of a function zero_ending(scores) to add all those values in the list of scores, which are ending with zero and display the sum.

Code:

```
def zero_ending(score):
    sum=0
    for i in score:
        if i % 10==0:
            sum+=i
    print("sum of all ending zero elements",sum)

l=[int(x) for x in input("Enter the elements of list \n").split()]
zero_ending(l)
```

```
ouput:
Enter the elements of list
10 20 25 30
sum of all ending zero elements 60
```

7.6 Write a definition of a method count_now(places) to find and display those place names,in which there are more than 5 characters.

```
Code:
def count_now(l):
    ap=[]
    for i in l:
        if len(i)>5:
            ap.append(i)
    print("place's naem having more than 5 character ",[x for x in ap])

l=[x for x in input("Enter the name of the city\n").split("\t")]
count_now(l)


ouput:
Enter the name of the city
Palastine       New York        Protugal        Ghana   Rome
place's naem having more than 5 character
 ['Palastine', 'New York', 'Protugal']
```

7.7 Write a method in python to display the elements of list thrice if it is a number and display the element terminated with '#' if it is not a number.

```
Code:
def modifylist(l):
    ls=[]
    for i in l:
        if i.isnumeric():
            ls.append(i*3)
        else:
            ls.append(i+"#")
    for x in ls:
        print(x)
ThisList=["41","DROND","GIRIRAJ", "13","ZARA"]
modifylist(ThisList)

ouput:
414141
DROND#
GIRIRAJ#
131313
ZARA#
```

7.8 For a given list of values in descending order, write a method in python to search for a value with the help of Binary Search method. The method should return position of the value and should return -1 if the value not present in the list.

```
Code:
def binarysearch(l,val):
    n=len(l)-1
    r=0
    while(r<=n):
        mid = (n+r)//2
        if l[mid]==val:
            return mid

        if l[mid]<val:
            r=mid+1
        else:
            n=mid-1
    return -1

ls=[81,64,51,28,23,0]
print("The list ",ls)
ls.sort()
print("sorting into asccending order",ls)
print("Enter the value to be searched")
val=int(input())
get=binarysearch(ls,val)
if  get==-1:
    print("return",get,"not in list")
else:
    print(val,"found at index ",get)

ouput:
Case - 1
The list  [81, 64, 51, 28, 23, 0]
sorting into asccending order [0, 23, 28, 51, 64, 81]
Enter the value to be searched
51
51 found at index  3
Case - 2
The list  [81, 64, 51, 28, 23, 0]
sorting into asccending order [0, 23, 28, 51, 64, 81]
Enter the value to be searched
100
return -1 not in list
```

7.9 Write a function half_and_half that takes in a list and change the list such that the elements of the second half are now in the first half.

```
Code:
def half_and_half(l):
    n=len(l)
    ml=list()
    if n%2==0:
        h=n//2
        ml+=(l[h:])
        ml+=(l[:h])
    else:
        h=n//2
        ml+=(l[h+1:])
        ml+=(l[:h])
        ml.insert(h,l[h])
    print(ml)

l=[10,20,30,40,50,60,70]
half_and_half(l)

l=[10,20,30,50,60,70]
half_and_half(l)

ouput:
[50, 60, 70, 40, 10, 20, 30]
[50, 60, 70, 10, 20, 30]
```

7.10Write a function that accepts a dictionary as an argument. If the dictionary contains duplicate values, it should return an empty dictionary. Otherwise, it should return a new dictionary where the values become the keys and the keys become the values.

Code:

```
def takedic(d):
    l=[]
    dic=dict()
    for i in d:
        if d[i] not in l:
           l.append(d[i])
    if len(l) != len(d):
        d.clear()
    else:
        dic=dict([(value,key) for (key,value) in d.items()])
    return dic
print("First case")

d1={
    'a':10,
    'b':20,
    'c':30
}
print(d1)
print(takedic(d1))
print("Second case")
d2={
    'a':10,
    'b':20,
    'c':20
}
print(d2)
print(takedic(d2))
```

ouput:

```
First case
{'a': 10, 'b': 20, 'c': 30}
{10: 'a', 20: 'b', 30: 'c'}
Second case
{'a': 10, 'b': 20, 'c': 20}
{}
```