

Mohona Das

1905756

TT Lab-3

1) Write a Python function to find the greatest among three numbers.

```
def maximum(a, b, c):  
    list = [a, b, c]  
    return max(list)  
a =(int) (input("Enter a no"))  
b =(int) (input("Enter a no"))  
c =(int) (input("Enter a no"))  
print(maximum(a, b, c))
```

```
Enter a no3  
Enter a no6  
Enter a no7  
7
```

2) Write a Python function to reverse a string

```
def reverse(string):  
    string = string[::-1]  
    return string  
  
s = input("Enter string")  
print ("original string : ",end="")  
print (s)  
  
print ("reversed string : ",end="")  
print (reverse(s))
```

```
Enter stringHello  
original string : Hello  
reversed string : olleH
```

3) Write a Python function that takes a list and returns a new list with unique elements of the first list. Sample List : [1,2,3,3,3,3,4,5] Unique List : [1, 2, 3, 4, 5]

```
def unique(list):  
    s=set(list)  
    return s
```

```
l=[1,2,2,2,3,3,3,5,6,7,2]
print ("original list : ",end="")
print (l)

print ("unique list: ",end="")
print (unique(l))

    original list : [1, 2, 2, 2, 3, 3, 3, 5, 6, 7, 2]
    unique list: {1, 2, 3, 5, 6, 7}
```

4) Write a Python function that checks whether a passed string is palindrome or not.

```
def is_palindrome(s):
    if len(s) < 1:
        return True
    else:
        if s[0] == s[-1]:
            return is_palindrome(s[1:-1])
        else:
            return False
a=str(input("Enter string:"))
if(is_palindrome(a)==True):
    print("Palindrome")
else:
    print("Not palindrome")

    Enter string:malayalam
    Palindrome
```

5) Write a Python function to find out the factorial of a number using recursion

```
def factorial(n):
    if n == 1:
        return n
    else:
        return n*factorial(n-1)

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

if n < 0:
    print("negative numbers: FACT NOT EXIST")
elif n == 0:
    print("The factorial of 0 is: 1")
else:
    print("The factorial of", n, "is :",factorial(n))

    Enter a number: 5
```

The factorial of 5 is : 120

6) Write a Python function to display the Fibonacci Series using recursion

```
def fibo(n):
    if n <= 1:
        return n
    else:
        return(fibo(n-1) + fibo(n-2))

nterms = int(input("Enter a n terms : "))

if nterms <= 0:
    print("NOT a positive integer")
else:
    print("Fibonacci sequence:")
    for i in range(nterms):
        print(fibo(i))

Enter a n terms : 6
Fibonacci sequence:
0
1
1
2
3
5
```

7) Write a Python function to find out Sum of Digits using recursion.

```
def sum_of_digit(n):
    if n < 10:
        return n
    else:
        return n%10 + sum_of_digit(n/10)

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

print("Sum of digit of number %d is %d." % (number,digit_sum))

Enter number: 120
Sum of digit of number 120 is 3.
```

8) Write a Python function to find out LCM and GCD using recursion.

```
def gcd(a,b):
    if (b == 0):
        return a;
    else:
        return gcd(b, a % b);
```

```
a=int(input("Enter a number : "))
b=int(input("Enter a number : "))
ans=gcd(a,b)
lcm=(a*b)/ans
print(ans)
print(lcm)
```

```
Enter a number : 5
Enter a number : 40
5
40.0
```

9) Write a Python function to implement Binary Search without using recursion.

```
def binary_search(arr, x):
    low = 0
    high = len(arr) - 1
    mid = 0
    while low <= high:
        mid = (high + low) // 2
        if arr[mid] < x:
            low = mid + 1
        elif arr[mid] > x:
            high = mid - 1
        else:
            return mid
    return -1
arr = [ 2, 3, 4, 10,23,27,33,39, 40 ]
x = int(input('enter the element to be searched: '))
result = binary_search(arr, x)
if result != -1:
    print("Element is present at index", str(result))
else:
    print("Element is not present")
```

```
enter the element to be searched: 4
Element is present at index 2
```

10) Write a Python function to implement Binary Search with using recursion.

```

def binary(a, f, l, s):
    mid=int((f+l)/2)
    if s>a[mid]:
        binary(a, mid, l, s)
    elif s<a[mid]:
        binary(a,f, mid, s)
    elif s==a[mid]:
        print("Number found at", mid+1)
    else:
        print("Number is not there in the array")
b=[1,2,3,4,5,6,7,8,9]
s = int(input('enter the element to be searched: '))
print("The list is ",b)
f=0
l=len(b)
binary(b,f,l,s)

```

```

enter the element to be searched: 6
The list is  [1, 2, 3, 4, 5, 6, 7, 8, 9]
Number found at 6

```

11). WAP to calculate the sum of digits of a given number.

```

n=int(input("Enter a number:"))
s=0
while(n>0):
    dig=n%10
    s+=dig
    n=n//10
print("The total sum of digits is:",s)

```

```

Enter a number:1345
The total sum of digits is: 13

```

12)WAP to check whether a number n is a prime number or not.

```

num =int(input("Enter number"))
if num > 1:
    for i in range(2,num):
        if (num % i) == 0:
            print(num,"is not a prime number")
            print(i,"times",num//i,"is",num)
            break
    else:
        print(num,"is a prime number")

```

```
else:
    print(num,"is not a prime number")
```

```
Enter number5
5 is a prime number
```

13)Write a function to check if a given number is perfect or not. The first perfect number is 6, because 1, 2,and 3 are its proper positive divisors, and  $1 + 2 + 3 = 6$  /\*Hints: A number is a perfect number if is equal to sum of its proper divisors, that is, sum of its positive divisors excluding the number itself.

```
n = int(input("Enter any number: "))
sum1 = 0
for i in range(1, n):
    if(n % i == 0):
        sum1 = sum1 + i
if (sum1 == n):
    print(" Perfect number")
else:
    print("Not a Perfect number")
```

```
Enter any number: 120
Not a Perfect number
```

14). WAP to print all odd and even numbers separately within a given range. The range is input through user.

```
start = int(input("Enter the start of range: "))
end = int(input("Enter the end of range: "))
e=list()
o=list()
for num in range(start, end + 1):
    if num % 2 != 0:
        o.append(num)
    else:
        e.append(num)
print(e,o)
```

```
Enter the start of range: 1
Enter the end of range: 4
[2, 4] [1, 3]
```

15). WAP to evaluate the equation  $y=x^n$  where n is a non-negative integer.

```
x=int(input("Enter base"))
n=int(input("Enter power"))
```

```

result = pow(x,n)

print("y=x^n=> " + str(result))

Enter base2
Enter power3
y=x^n=> 8

```

16). WAP to print the series as 1 2 7 15 31 .....n, where n is given by the user

```

n=int(input("Enter n "))
i = 1
p = 0

while i <= n:
    p= (p* 2) + 1
    print(p, end=" ")
    i += 1

```

```

Enter n 6
1 3 7 15 31 63

```

17). WAP to print the series as 3 5 7 11 13 17.....n, where n is given by the user.

```

def prime(x, y):
    prime_list = []
    for i in range(x, y):
        if i == 0 or i == 1:
            continue
        else:
            for j in range(2, int(i/2)+1):
                if i % j == 0:
                    break
            else:
                prime_list.append(i)
    return prime_list
ending_range =int(input("Enter n"))
lst = prime(3, ending_range)
if len(lst) == 0:
    print("There are no prime numbers in this range")
else:
    print("The prime numbers in this range are: ", lst)

```

```

Enter n22
The prime numbers in this range are: [3, 5, 7, 11, 13, 17, 19]

```

18). WAP to sum the following series  $S=1+(1+2)+(1+2+3)+\dots+(1+2+3+\dots+n)$

```
def sumOfSeries(n):
    return sum([i*(i+1)/2 for i in range(1, n + 1)])
```

```
n = int(input("Enter n "))
print(sumOfSeries(n))
```

```
Enter n 3
10.0
```

19). WAP to print the following pattern for n rows. Ex. for n=5 rows

```
•
•  ○
•  ○  ■
•  ○  ■  ■
•  ○  ■  ■  ■
```

```
def pat(n):
    for i in range(1, n + 1):
        for j in range(1, i + 1):
            print("*", end=' ')
        print('')
```

```
n = int(input("Enter n "))
pat(n)
```

```
Enter n 5
*
* *
* * *
* * * *
* * * * *
```

20). WAP to print the following pattern for n rows. Ex. for n=5 rows

```
1
2 1
1 2 3
```



4 3 2 1

```
def pat(n):
    for i in range(1, n+1):
        if i%2 == 0 :
            for j in range(i, 0, -1):
                print(j, end=' ')
            print("")
        else :
            for j in range(1,i+1,1):
                print(j,end=' ')
            print("")
```

```
n = int(input("Enter n "))
pat(n)
```

Enter n 5

1

2 1

1 2 3

4 3 2 1

1 2 3 4 5

21). WAP to convert a decimal number into its equivalent number with base b. Decimal number and b are the user input.

```
def reVal(num):
    if (num >= 0 and num <= 9):
        return chr(num + ord('0'))
    else:
        return chr(num - 10 + ord('A'))
def strev(str):
    l = len(str)
    for i in range(int(l / 2)):
        temp = str[i]
        str[i] = str[l - i - 1]
        str[l - i - 1] = temp
def fromDeci(res, base, inputNum):
    index = 0
    while (inputNum > 0):
        res+= reVal(inputNum % base)
        inputNum = int(inputNum / base)
    res = res[::-1]
    return res
```

```
inputNum = int(input("Enter n "));
base = int(input("Enter base "));
res = "";
print("Equivalent of", inputNum, "in base",base, "is", fromDeci(res, base, inputNum));
```

```

Enter n 4
Enter base 16
Equivalent of 4 in base 16 is 4

```

22). WAP to convert a number with base b into its equivalent decimal number. Numbers with base b & b are the user input.

```

def val(c):
    if c >= '0' and c <= '9':
        return ord(c) - ord('0')
    else:
        return ord(c) - ord('A') + 10;
def toDeci(str,base):
    llen = len(str)
    power = 1 #Initialize power of base
    num = 0 #Initialize result
    for i in range(llen - 1, -1, -1):
        if val(str[i]) >= base:
            print('Invalid Number')
            return -1
        num += val(str[i]) * power
        power = power * base
    return num
strr = input("Enter number ")
base = int(input("Enter base "))
print('Decimal equivalent of', strr,
      'in base', base, 'is',
      toDeci(strr, base))

```

```

Enter number A
Enter base 16
Decimal equivalent of A in base 16 is 10

```

23). WAP to convert a binary number to its equivalent octal- hexa-decimal number system.

```

dec = int(input("Enter n "))

print("The decimal value of", dec, "is:")
print(bin(dec), "in binary.")
print(oct(dec), "in octal.")
print(hex(dec), "in hexadecimal.")

```

```

Enter n 10
The decimal value of 10 is:

```

```
0b1010 in binary.  
0o12 in octal.  
0xa in hexadecimal.
```

24) write a python function to implement ( take user input list to display output)

- a) Bubble sort()
- b) Selection sort()
- c) Linear Search()
- d) Quick sort()
- e) Insertion Sort()

```
def bubbleSort(array):  
    for i in range(len(array)):  
        for j in range(0, len(array) - i - 1):  
            if array[j] > array[j + 1]:  
                temp = array[j]  
                array[j] = array[j+1]  
                array[j+1] = temp
```

```
    print(array)
```

```
def selectionSort(array, size):  
    for step in range(size):  
        min_idx = step  
        for i in range(step + 1, size):  
            if array[i] < array[min_idx]:  
                min_idx = i  
        (array[step], array[min_idx]) = (array[min_idx], array[step])  
    print(array)
```

```
def partition(array, low, high):  
    pivot = array[high]  
    i = low - 1  
    for j in range(low, high):  
        if array[j] <= pivot:  
            i = i + 1  
            (array[i], array[j]) = (array[j], array[i])  
    (array[i + 1], array[high]) = (array[high], array[i + 1])  
    return i + 1
```

```
def quickSort(array, low, high):  
    if low < high:  
        pi = partition(array, low, high)  
        quickSort(array, low, pi - 1)  
        quickSort(array, pi + 1, high)
```

```
def insertionSort(array):
```

```

    for step in range(1, len(array)):
        key = array[step]
        j = step - 1
        while j >= 0 and key < array[j]:
            array[j + 1] = array[j]
            j = j - 1
        array[j + 1] = key
    print(array)

data = [9, 5, 1, 4, 3,3,6,43,12,54]

def linearSearch(array, n, x):
    for i in range(0, n):
        if (array[i] == x):
            return i
    return -1

x = 1
n = len(data)
result = linearSearch(data, n, x)
if(result == -1):
    print("Element not found")
else:
    print("Element found at index: ", result)

print("bubble sort " , bubbleSort(data))
print("Selection sort ", selectionSort(data,len(data)))
print("Insertion sort ",insertionSort(data))

```

```

Element found at index: 2
[1, 3, 3, 4, 5, 6, 9, 12, 43, 54]
bubble sort  None
[1, 3, 3, 4, 5, 6, 9, 12, 43, 54]
Selection sort  None
[1, 3, 3, 4, 5, 6, 9, 12, 43, 54]
Insertion sort  None

```

25) write function to find smallest number in a list

```

def mini(l):
    #for i in range(0, len(l)):
        return min(l)
l=[1,2,4,5,6,7,32,-9,23,155,6545]
print(mini(l))

```

26) write a python function to find largest number in a list

```
def maxi(l):
    #for i in range(0, len(l)):
        return max(l)
l=[1,2,4,5,6,7,32,-9,23,155,6545]
print(maxi(l))

6545
```

27) write a python function to find second largest number in a list

```
def maxi(l):
    m=[x for x in l if x !=max(l)]
    return max(m)

l=[1,2,4,5,6,7,32,-9,23,155,6545]
print(maxi(l))

155
```

28) write a lambda function to implement :

- a) Square of number (use map function to find)
- b) Cube of number
- c) Max of two number (4,5)

```
nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print(nums)
sq = list(map(lambda x: x ** 2, nums))
print(sq)
cb = list(map(lambda x: x ** 3, nums))
print(cb)

a=int(input("Enter a number: "))
b=int(input("Enter a number: "))
maximum = lambda a,b:a if a > b else b
print(f'{maximum(a,b)} is a max')

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
[1, 8, 27, 64, 125, 216, 343, 512, 729, 1000]
Enter a number: 56
Enter a number: 24
56 is a max
```

29) Write a python program to find the square and cube of the list L: [1,2,3,4,5,6,7,8,9,10] Using map function

```
def sq(n):
    return n * n

def cb(n):
    return n*n*n
nums = [4, 5, 2, 9]
print("Original List: ",nums)
result = map(sq, nums)
print("Square of elements ")
print(list(result))
result = map(cb, nums)
print("Cube of elements ")
print(list(result))

Original List: [4, 5, 2, 9]
Square of elements
[16, 25, 4, 81]
Cube of elements
[64, 125, 8, 729]
```

30) Write a python program to number which are prime numbers in the given list L: [1,2,3,4,5,6,7,8,9,10] Using filter function.

```
def fun(num):
    if num > 1:
        for i in range(2,num):
            if (num % i) == 0:
                return False
            break
        else:
            return True
    else:
        return False
l=[1,2,3,4,5,6,7,8,9,10]

# using filter function
filtered = filter(fun, l)

print('The Prime nos. are:')
for s in filtered:
    print(s)
```

The Prime nos. are:  
3

5  
7  
9

31) Write a python program to find following operations for the give the list L: [1,2,3,4,5,6,7,8,9,10]  
Using reduce function a. compute maximum element from list b. compute sum of list

```
import functools
lis = [1, 3, 5, 6, 2,4,5,6,865,312,231 ]
print("The sum of the list elements is : ", end="")
print(functools.reduce(lambda a, b: a+b, lis))
print("The maximum element of the list is : ", end="")
print(functools.reduce(lambda a, b: a if a > b else b, lis))
```

```
The sum of the list elements is : 1440
The maximum element of the list is : 865
```

32) Given list of elements, the task here is to write a Python function that can remove presence of all a specific digit from every element and then return the resultant list Input : test\_list = [333, 893, 1948, 34, 2346], K = 3 Output : [89, 1948, 4, 246] Explanation : All occurrences of 3 are removed

```
test_list = [345, 893, 1948, 34, 2346]
print("The original list is : " + str(test_list))
K = 3
res = []
for ele in test_list:
    if list(set(str(ele)))[0] == str(K) and len(set(str(ele))) == 1:
        res.append('')
    else:
        res.append(int(''.join([el for el in str(ele) if int(el) != K])))
print("Modified List : " + str(res))
```

```
The original list is : [345, 893, 1948, 34, 2346]
Modified List : [45, 89, 1948, 4, 246]
```

---

✓ 3s completed at 6:50 PM



Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.