Final Exam

Sol 1:

```
table_number= int(input("Enter the table number you want to print: "))

for n in range(1, 11):
    print(table_number, "*", n, "=", table_number*n)
```

Sol 2:

```
import math

def haversine(long1, long2, lat1, lat2):
    long1= radians(long1)
    long2= radians(long2)
    lat1= radians(lat1)
    lat2= radians(lat2)

    dlong= long2 - long1
    dlat= lat2 - lat1

    a= sin * (dlat/2)**2 + cos * (lat1) * cos(lat2) * sin * (dlong/2)**2
    c= 2 * asin (math.sqrt(a))
    r= 6371
    distance= r * c
    print(distance)
```

Sol 3:

A)

```
n= 6

for i in range(0, n):
    for j in range(i):
        print("*", end = " ")
    print()
```

B)

```
n= 6

for i in range(n, 0, -1):
    for j in range(i):
        print("*", end = " ")
    print()
```

Sol 4:

```
from tkinter import *

def btnClick(numbers):
    global operator
    operator= operator + str(numbers)
    text_Input.set(operator)

def clear_display():
    global operator
    operator= " "
    text_Input.set("")

def equality_function():
    global operator
    result= str(eval(operator))
    text_Input.set(result)
    operator= " "


calculator= Tk()
calculator.title("Simple Calculator")

operator = " "

text_Input = StringVar()

textDisplay = Entry(calculator, font=("arial",22, "bold"), textvariable=
text_Input, bd=25, bg= "powder blue", insertwidth=4, justify=
"right").grid(columnspan=4)

btn7= Button(calculator, padx=18, font=("arial",22,"bold"), fg="blue",
bg="white", text= "7", command= lambda:btnClick(7)).grid(row=1, column=0)
btn8= Button(calculator, padx=18, font=("arial",22,"bold"), fg="blue",
bg="white", text= "8", command= lambda:btnClick(8)).grid(row=1, column=1)
btn9= Button(calculator, padx=18, font=("arial",22,"bold"), fg="blue",
bg="white", text= "9", command= lambda:btnClick(9)).grid(row=1, column=2)
ADDITION= Button(calculator, padx=18, font=("arial",22,"bold"), fg="blue",
bg="white", text= "+", command= lambda:btnClick("+")).grid(row=1, column=3)

btn4= Button(calculator, padx=18, font=("arial",22,"bold"), fg="blue",
bg="white", text= "4", command= lambda:btnClick(4)).grid(row=2, column=0)
btn5= Button(calculator, padx=18, font=("arial",22,"bold"), fg="blue",
bg="white", text= "5", command= lambda:btnClick(5)).grid(row=2, column=1)
btn6= Button(calculator, padx=18, font=("arial",22,"bold"), fg="blue",
bg="white", text= "6", command= lambda:btnClick(6)).grid(row=2, column=2)
```

```
SUBTRACTION= Button(calculator, padx=18, font=("arial",22,"bold"), fg="blue",
bg="white", text= "-", command= lambda:btnClick("-")).grid(row=2, column=3)

btn1= Button(calculator, padx=18, font=("arial",22,"bold"), fg="blue",
bg="white", text= "1", command= lambda:btnClick(1)).grid(row=3, column=0)
btn2= Button(calculator, padx=18, font=("arial",22,"bold"), fg="blue",
bg="white", text= "2", command= lambda:btnClick(2)).grid(row=3, column=1)
btn3= Button(calculator, padx=18, font=("arial",22,"bold"), fg="blue",
bg="white", text= "3", command= lambda:btnClick(3)).grid(row=3, column=2)
MULTIPLE= Button(calculator, padx=18, font=("arial",22,"bold"), fg="blue",
bg="white", text= "x", command= lambda:btnClick("*")).grid(row=3, column=3)

btnclean= Button(calculator, padx=18, font=("castellar",15,"italic"),
fg="black", bg="white", text= "CLR", command= clear_display).grid(row=4,
column=0)
btn0= Button(calculator, padx=18, font=("arial",22,"bold"), fg="blue",
bg="white", text= "0", command= lambda:btnClick(0)).grid(row=4, column=1)
btnequal= Button(calculator, padx=18, font=("arial black",22,"bold"),
fg="blue", bg="white", text= "=", command= equality_function).grid(row=4,
column=2)
DIVISION= Button(calculator, padx=18, font=("arial black",22,"bold"),
fg="blue", bg="white", text= "/", command= lambda:btnClick("/")).grid(row=4,
column=3)

calculator.mainloop()
```

Sol 5:

- ***Lambda Functions:***

  They are essential if you're using some functions in the program only once. They are also called single-line functions as well as anonymous functions.

Code for a) and b):

```
list1= [2,4,5,7,12,18]
list2= [1,14,58,17,2,19]

a= filter(lambda x: x % 2 == 0, list1) #even numbers remainder is always 0,
#list1 is added to tell the directory, #filter is a variable used to filter
the result
b= filter(lambda y: y % 2 == 0, list2)

print(list(a))
print(list(b)) #list() is use here to tell the form of printing of result
```