NAME: M ZOHAB AHMER
REG NO. : FA20-BCS-038
SEM/SEC : 5/B
SUBJECT : AI LAB
DATED : 30-09-2022

LAB ASSIGNMENT

Q:-Create two lists based on the user values. Merge both the lists and display in sorted order.

```
a=[]
c=[]
n1=int(input("Enter number of elements:"))for i in range(1,n1+1):
    b=int(input("Enter element:"))
    a.append(b)
n2=int(input("Enter number of elements:"))for i in range(1,n2+1):
    d=int(input("Enter element:"))
    c.append(d)new=a+cnew.sort()print("Sorted list is:",new)
```

Lab Task 2:
Repeat the above activity to find the smallest and largest element of the list. (Suppose all the elements are integer values)

```
# Python 3 program to print
# the array in given order

# Function which arrange the
# array.
def rearrangeArray(arr, n) :

        # Sorting the array elements
        arr.sort()

        # To store modified array
        tempArr = [0] * (n + 1)

        # Adding numbers from sorted
        # array to new array accordingly
        ArrIndex = 0

        # Traverse from begin and end
        # simultaneously
        i = 0
        j = n-1
```

```
    while(i <= n // 2 or j > n // 2 ) :

        tempArr[ArrIndex] = arr[i]
        ArrIndex = ArrIndex + 1
        tempArr[ArrIndex] = arr[j]
        ArrIndex = ArrIndex + 1
        i = i + 1
        j = j - 1

    # Modifying original array
    for i in range(0, n) :
        arr[i] = tempArr[i]


# Driver Code
arr = [ 5, 8, 1, 4, 2, 9, 3, 7, 6 ]
n = len(arr)
rearrangeArray(arr, n)

for i in range(0, n) :
    print( arr[i], end = " ")

# This code is contributed by Nikita Tiwari.
```

Lab Task 3:

The derivate of a function f(x) is a measurement of how quickly the function f changes with respect to
change in its domain x. This measurement can be approximated by the following relation,

Sol:

```
def
show_all_in_dict(dict):
                        """
                        Print all the key in dict.

                        Arguments:
                        dict -- a dictionary needed to be print it's key.

                        """
                        print('We know the birthdays of:')
                        for key in dict:
                        print(key)

                        # Useless
                        def search(query, dict):
                        """
                        Search the person's birthday is in the dict or not.

                        Arguments:
                        query -- the name of person the user want to know.
                        dict -- a dictionary needed to be print it's key.

                        Returns:
                        birthday -- the birthday of the query.
                        """
```

```
return dict[query] if query in dict else None

def main():
Birthdays ={"Albert Einstein": "14/3/1889",
"Bill Gates": "28/10/1955",
"Steve Jobs": "24/2/1955"}
print('Welcome to the birthday dictionary.')
show_all_in_dict(Birthdays)
query = input("Who's birthday do you want to look up?")
result = Birthdays[query] if query in Birthdays else None
if result == None:
print('No Data')
else:
print("{}'s birthday is {}".format(query, Birthdays[query]))


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

Create a dictionary by extracting the keys from a given dictionary

Write a Python program to create a new dictionary by extracting the mentioned keys from the below dictionary.

```
employees = ['Kelly', 'Emma']
defaults = {"designation": 'Developer', "salary": 8000}
res = dict.fromkeys(employees, defaults)
 print(res)
 # Individual data
print(res["Kelly"])
```