

# Lab1 Python Programming

Mishan Regmi 5BCA A

1841030

Code:

```
#A.Implement the sequential search algorithm in Python.
```

```
#B. Create a list of covid affected countries sort it in a reverse order and store it into a new list.
```

```
# Using linear search Check the position of India and display it.
```

```
#C. Ternary search (embed in any domain)
```

```
#NOTES: C. is optional
```

```
print("Lab 1 Python")
```

```
print("Mishan Regmi 5BCA A '1841030'")
```

```
print("\nSequential Searching")
```

```
print("\nSystem to check employee attendance.\n")
```

```
name = input("Enter the name of Employee:")
```

```
empl = ['Rakesh', 'Hari', 'Shyam', 'Krishna', 'Shova', 'Rekha', 'Prasuram', 'Jai', 'Mishan', 'Sushma']
```

```
def sequential(employee, item):
```

```
    position = 0
```

```
    pos = "Employee is on duty today."
```

```
    neg = "Employee was on leave."
```

```

        while position < len(employee):
            if employee[position] == item:
                return pos
            position = position + 1
        return neg

print(sequential(empl, name))

print("\nLinear Seach")
countryname = ['USA', 'Brazil', 'Russia', 'Spain', 'U
K', 'Italy', 'France', 'Germany', 'India', 'Turkey']
revLst = countryname[::-1]

def linear(con, x):
    notfnd= "Country Not Found"
    for i in range(len(con)):
        if con[i] == x:
            return i
    return notfnd
ser = 'India'
print("India Found At: "+str(linear(revLst,ser)))

print("\nTernary Search")

# l = starting index
# r = length of our list
# key = element to be searched
# ar = name of the list

def ternarySearch(l, r, key, ar):

```

```

    if (r >= 1): #confirming the length of list is correct
        mid1 = 1 + (r - 1) //3 # looking for mid1
        mid2 = r - (r - 1) //3 # looking for mid2
        if (ar[mid1] == key): #checking match with mid 1
            return mid1
        if (ar[mid2] == key): #checking match with mid 2
            return mid2
        if (key < ar[mid1]): #if key is less than mid 1
            return ternarySearch(1, mid1 - 1, key, ar) #another iteration and search on before mid 1
        return ternarySearch(mid2 + 1, r, key, ar) #another iteration and search on after mid 1
    else:
        return ternarySearch(mid1 + 1, mid2 - 1, key, ar) #search in mid 2
    return -1
#initialize of value for the search
ar = [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ] #list
l = 0 #starting index
r = 9 #end index
key = 7 #value to be searched
p = ternarySearch(l, r, key, ar) #the function
print("Index of", key, "is", p)

```