Lab 4

Task 1.

Binary Search

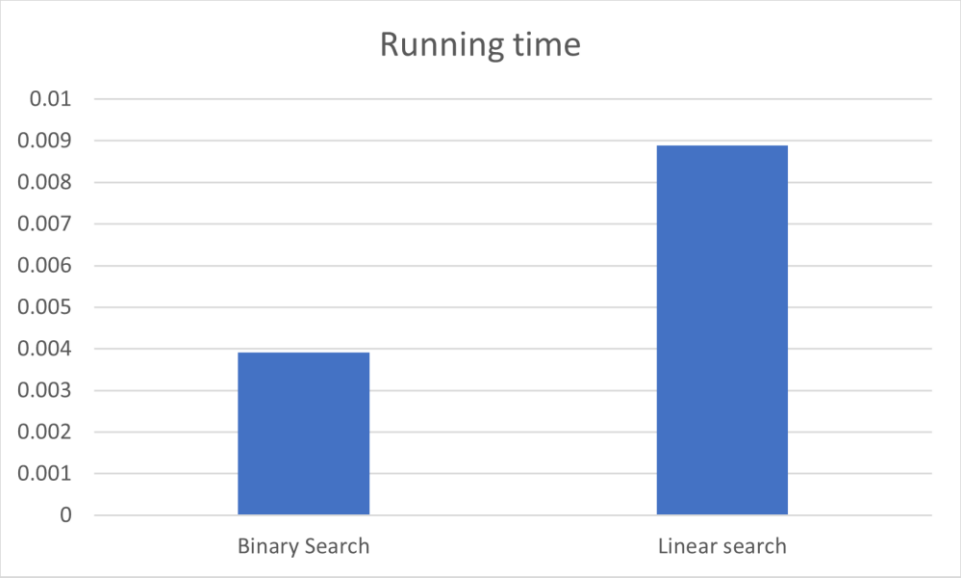
```
import time


def binarySearch(arr, l, r, x):
    while l <= r:

        mid = l + (r - l) // 2;

        if arr[mid] == x:
            return mid

        # If x is greater, ignore left half
        elif arr[mid] < x:
            l = mid + 1

        # If x is smaller, ignore right half
        else:
            r = mid - 1

    # If we reach here, then the element
    # was not present
    return -1


start = time.time()
# Driver Code
arr = [i for i in range(100000)]
x = 9999

# Function call
result = binarySearch(arr, 0, len(arr) - 1, x)

if result != -1:
    print("Element is present at index % d" % result)
    print("time taken for ", input, ":", time.time() - start)
else:
    print("Element is not present in array")
```

Linear Search

```
import time

def search(arr, n, x):
    for i in range(0, n):
        if (arr[i] == x):
            return 1
    return -1;

start = time.time()

arr = [ i for i in range(100000)]
x = 9999
```

```
n = len(arr)

result = search(arr, n, x)
if result == -1:
    print("Element is no in the array")
else:
    print("Element is found at index", result)
    print("time taken for ", input, ":", time.time() - start)
```

Comparison between linear and binary search running time.



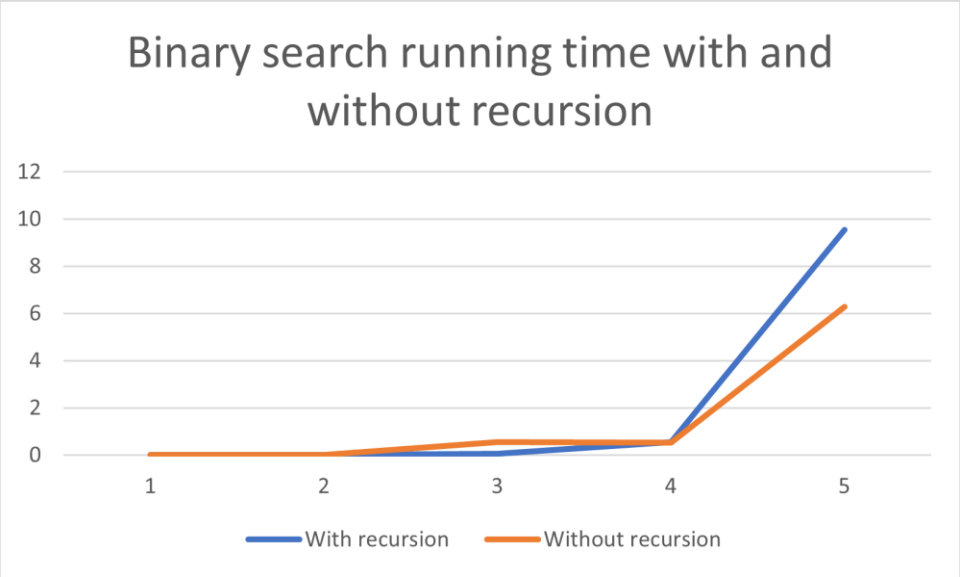
Task 2.

Iterative binary search, without recursion.

```
import time


def binary_search(arr, x):
    l = 0
    r = len(arr) - 1
    mid = 0

    while l <= r:

        mid = (r + l) // 2

        # If x is greater, ignore left half
        if arr[mid] < x:
            l = mid + 1

        # If x is smaller, ignore right half
        elif arr[mid] > x:
            r = mid - 1
```

```
        # means x is present at mid
        else:
            return mid

    # If we reach here, then the element was not present
    return -1


start = time.time()

# Test array
arr = [i for i in range(100000)]
x = 9999

# Function call
result = binary_search(arr, x)

if result != -1:
    print("Element is present at index", str(result))
    print("time taken for ", input, ":", time.time() - start)
else:
    print("Element is not present in array")
```

Comparison between running time in binary search with and without recursion. As the numbers increased, running time was faster without using recursion.

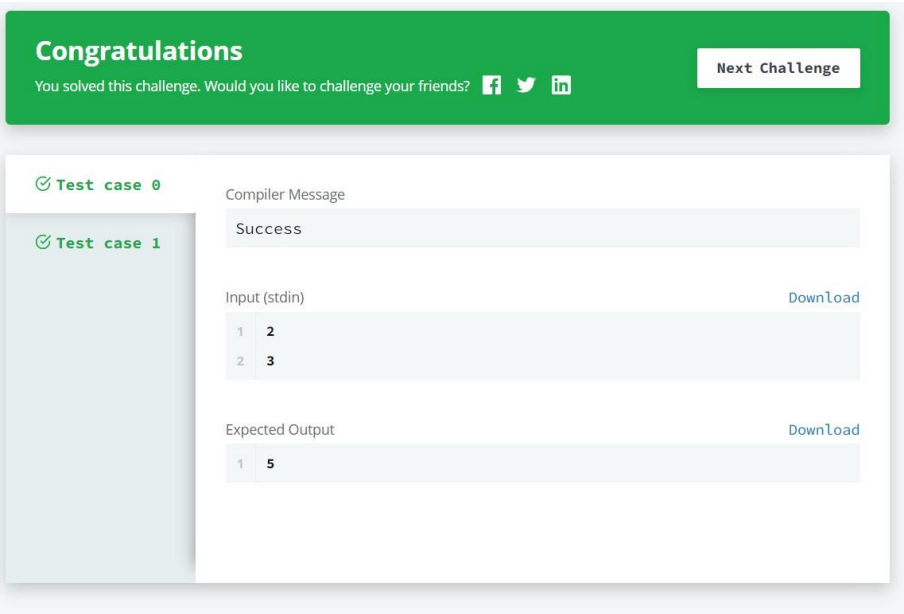


Task 3.

Signing up in Hacker Rank.

Task 4.

Completing the first task in Hacker Rank.



Task 5.

```
#!/bin/python3

import math
import os
import random
import re
import sys

n = int(input())
m = input().strip().split(' ')
res = m[::-1]
r=""

for i in range(n):
    r = r + str(res[i]) + " "

print(r)

```