

**ΑΣΚΗΣΗ 1η**

Να μελετηθεί και να εκτελεστεί ο παρακάτω κώδικας που υλοποιεί την **γραμμαμική αναζήτηση**.

```
lst = []
num = int(input("Enter size of list: \t"))
for n in range(num):
    numbers = int(input("Enter any number: \t"))
    lst.append(numbers)

x = int(input("\nEnter number to search: \t"))

found = False

for i in range(len(lst)):
    if lst[i] == x:
        found = True
        print("\n%d found at position %d" % (x, i+1))
        break
if not found:
    print("\n%d is not in list" % x)
```

**ΑΣΚΗΣΗ 2η**

Να μελετηθεί και να εκτελεστεί ο παρακάτω κώδικας που υλοποιεί την **δυναμική αναζήτηση**.

```
# Python code to implement iterative Binary Search.
# It returns location of x in given array arr
# if present, else returns -1
#arr:array, l:starting index, r:last_index, x:element
def binarySearch(arr, l, r, x):
    while l <= r:

        mid =int( (l + r)/ 2)

        # Check if x is present at mid
        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

# Test array-already sorted
arr = [2,14,19,21,99,210,512,1028,4443,5110, 5111, 5112, 5113, 5114, 5115, 5116,
5117, 5118, 5119, 5200]
x = 5200

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

if result != -1:
    print ("Element is present at index %d" % result)
else:
    print ("Element is not present in array")
```

**ΑΣΚΗΣΗ 3η**

Να μελετηθεί και να εκτελεστεί ο παρακάτω κώδικας που υλοποιεί την **αναζήτηση παρεμβολής**.

```
def intpolsearch(values,x):
    idx0 = 0
    idxn = (len(values) - 1)

    while idx0 <= idxn and x >= values[idx0] and x <= values[idxn]:
        # Find the mid point
        mid = idx0 + \
            int(((float(idxn - idx0)/( values[idxn] - values[idx0]))
                * ( x - values[idx0])))

        # Compare the value at mid point with search value
        if values[mid] == x:
            return "Found "+str(x)+" at index "+str(mid)
        if values[mid] < x:
            idx0 = mid + 1
    return "Searched element not in the list"

Data = [7, 9, 19, 21, 37, 38, 54, 85, 111]
print(intpolsearch(Data, 19))
```

**ΑΣΚΗΣΗ 4η**

Να μελετηθεί και να εκτελεστεί ο παρακάτω κώδικας που υλοποιεί την **αναζήτηση άλματος**.

```
import math

def JumpSearch (values, x):
    length = len(values)
    jump = int(math.sqrt(length))
    left, right = 0, 0
    while left < length and values[left] <= x:
        right = min(length - 1, left + jump)
        if values[left] <= x and values[right] >= x:
            break
        left += jump;
    if left >= length or values[left] > x:
        return -1
    right = min(length - 1, right)
    i = left
    while i <= right and values[i] <= x:
        if values[i] == x:
            return i
        i += 1
    return -1

Data = [7, 9, 19, 21, 37, 38, 54, 85, 111]
print(JumpSearch(Data, 19))
```