```
def binarysearch(roll, x):
    low = 0
    high = len(roll) - 1

    while high >= low:
        mid = (low + high) // 2

        if roll[mid] == x:
            return mid
            break
        elif roll[mid] > x:
            high = mid - 1
        elif roll[mid] < x:
            low = mid + 1
    return -1

def fibonacci_search(arr, n, key):
    offset = -1
    Fm2 = 0
    Fm1 = 1
    Fm = Fm2 + Fm1
    while Fm < n:
        Fm2 = Fm1
        Fm1 = Fm
        Fm = Fm2 + Fm1
    while Fm > 1:
        i = min(offset + Fm2, n - 1)
        if arr[i] < key:
            Fm = Fm1
            Fm1 = Fm2
            Fm2 = Fm - Fm1
            offset = i
        elif arr[i] > key:
            Fm = Fm2
            Fm1 = Fm1 - Fm2
            Fm2 = Fm - Fm1
        else:
            return i
    if Fm1 == 1 and arr[n - 1] == key:
        return n - 1
    return -1
```

i

```
roll_no = [2, 3, 4, 5, 6, 7, 8, 9]
n = len(roll_no)
key = int(input("enter the roll no to check whether he or she attended the training program"))
while True:
    print("1.Binary Search\n2.Fibonacci Search\n3.Exit")
    ch = int(input("Enter the choice"))
    if ch == 1:
        result = binarysearch(roll_no, key)
        if result == -1:
            print("Roll no : ", key, " not attended training program")
        else:
            print("Roll no : ", key, " attended training program")
    elif ch == 2:
        result = fibonacci_search(arr, n, key)
        if result >=0:

            print("Roll no : ", key, " attended training program")
        Else:
            print("Roll no : ", key, " not attended training program")

    elif ch == 3:
        break
    else:
        print("Enter valid choice")
```