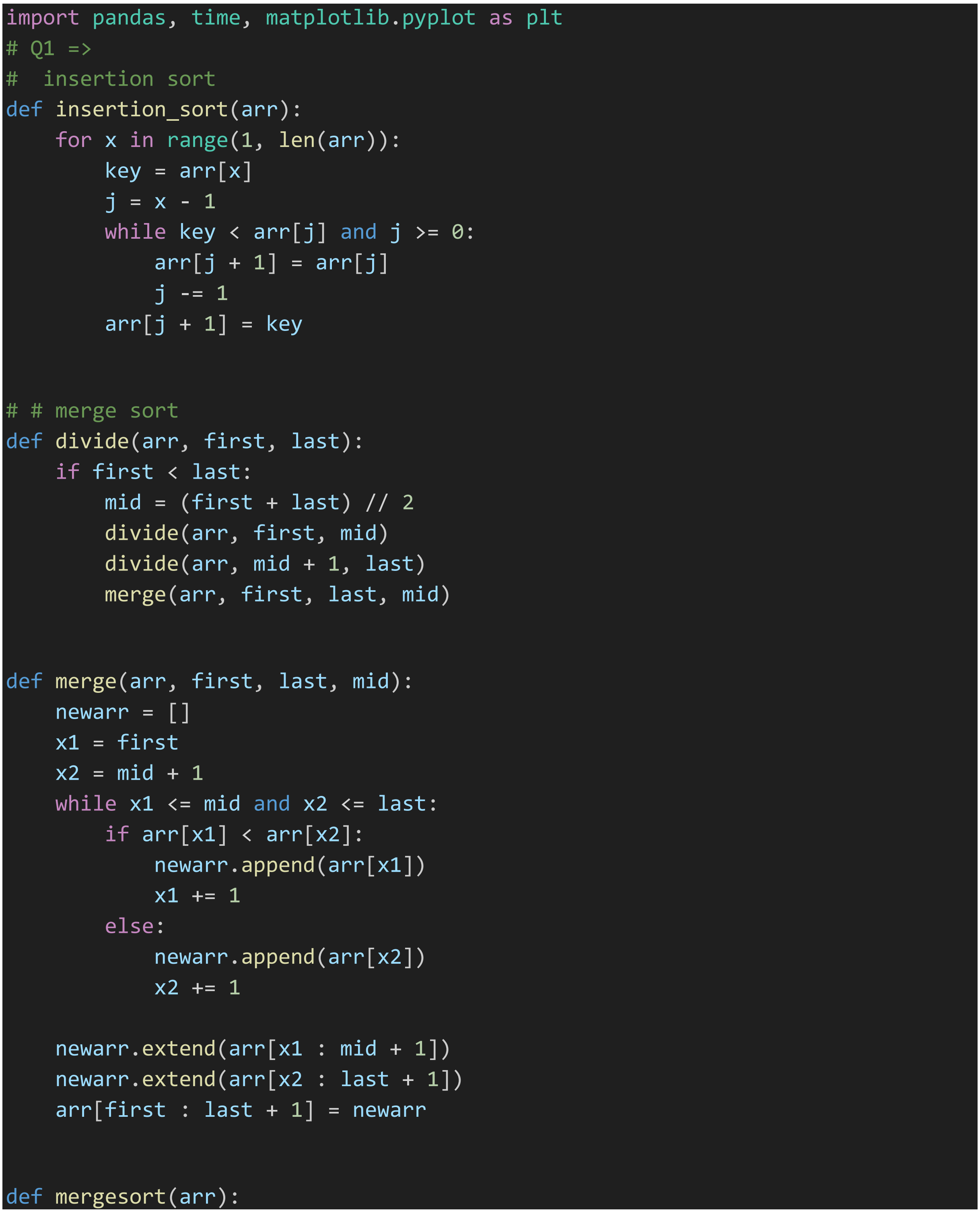
Practical No 2

Name : Sujal Nimje

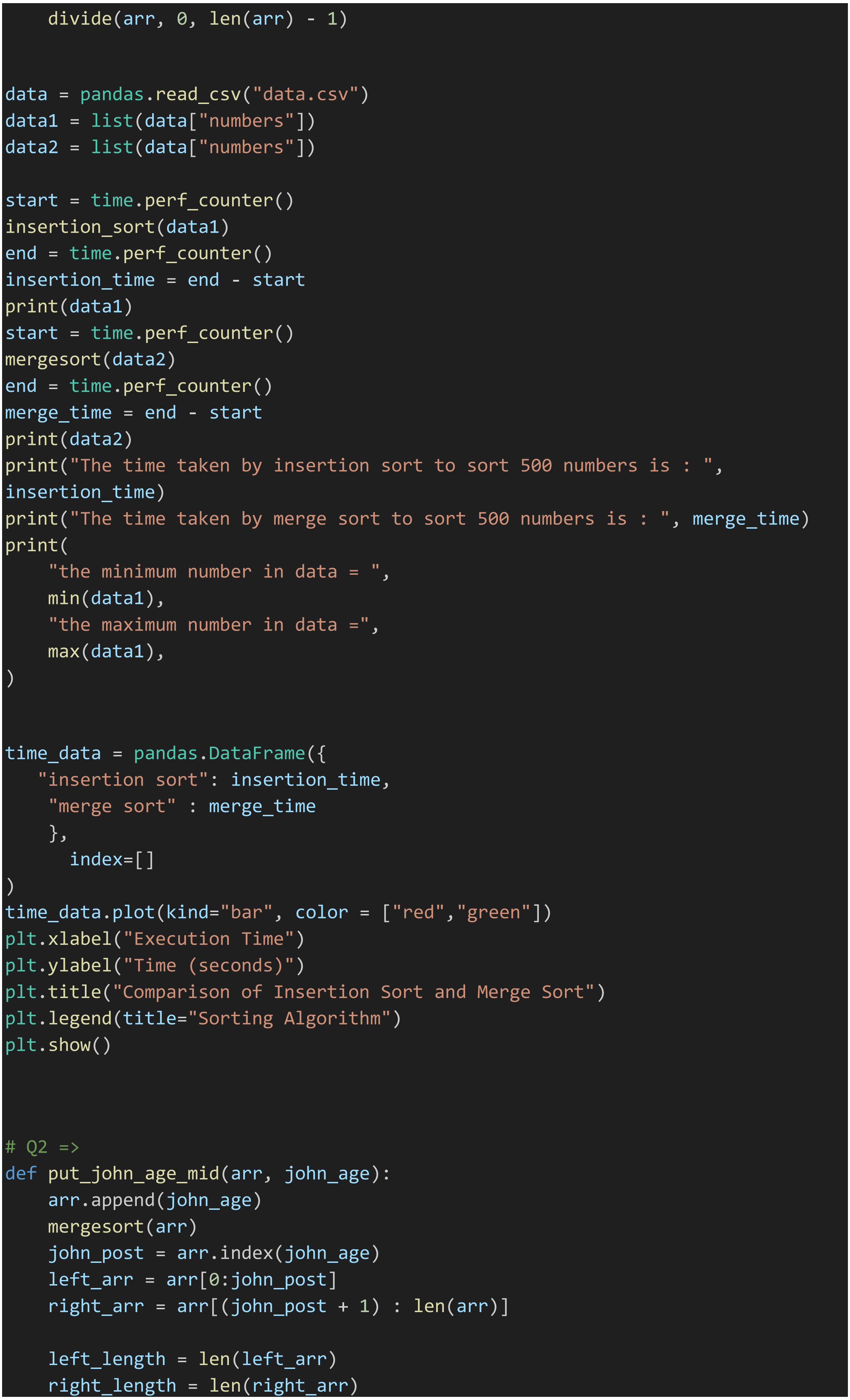
Roll No : 64

Batch : A4

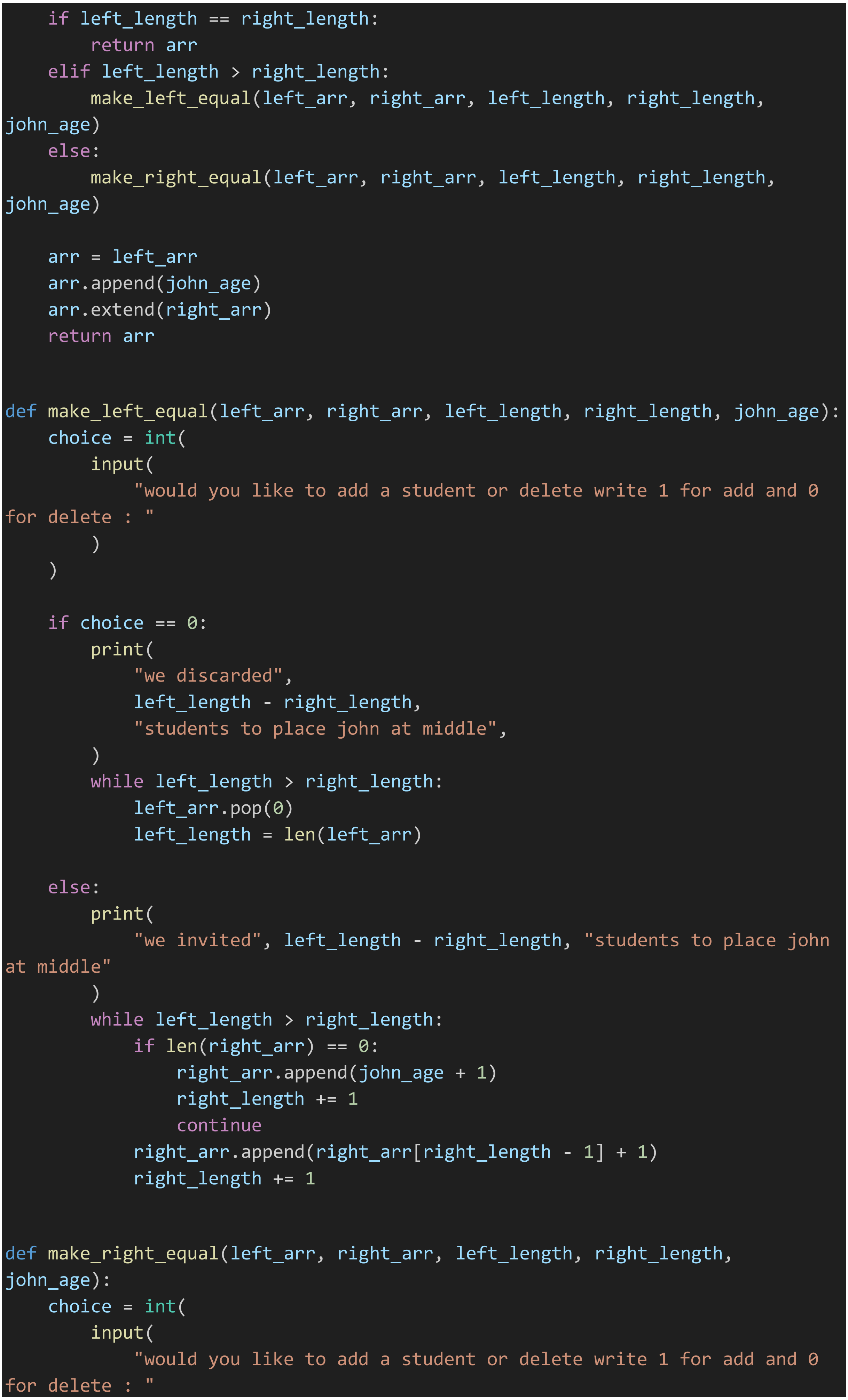
Code :



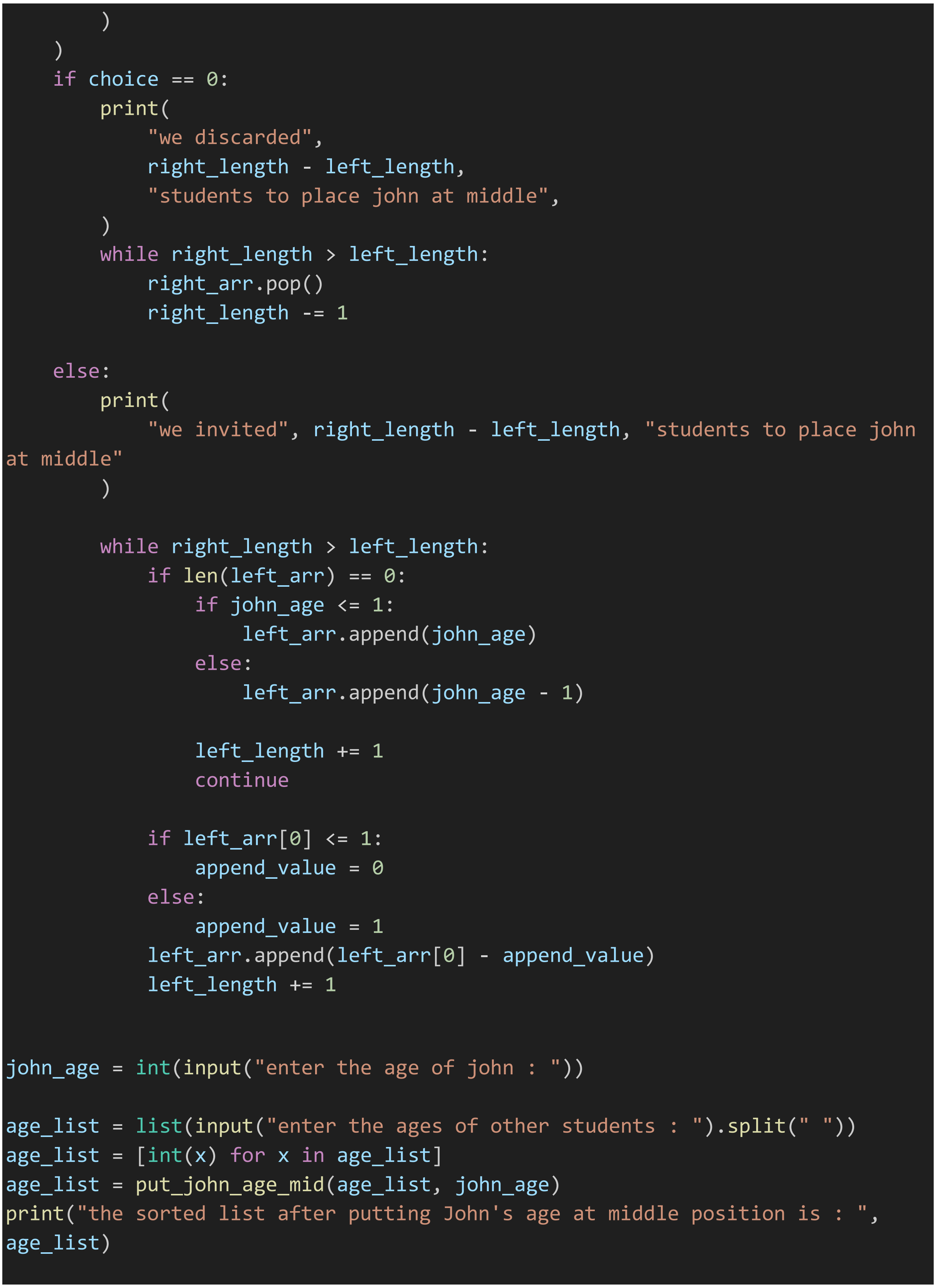
```
import pandas, time, matplotlib.pyplot as plt
# Q1 =>
#  insertion sort
def insertion_sort(arr):
    for x in range(1, len(arr)):
        key = arr[x]
        j = x - 1
        while key < arr[j] and j >= 0:
            arr[j + 1] = arr[j]
            j -= 1
        arr[j + 1] = key


# # merge sort
def divide(arr, first, last):
    if first < last:
        mid = (first + last) // 2
        divide(arr, first, mid)
        divide(arr, mid + 1, last)
        merge(arr, first, last, mid)


def merge(arr, first, last, mid):
    newarr = []
    x1 = first
    x2 = mid + 1
    while x1 <= mid and x2 <= last:
        if arr[x1] < arr[x2]:
            newarr.append(arr[x1])
            x1 += 1
        else:
            newarr.append(arr[x2])
            x2 += 1

    newarr.extend(arr[x1 : mid + 1])
    newarr.extend(arr[x2 : last + 1])
    arr[first : last + 1] = newarr


def mergesort(arr):
```



```
    divide(arr, 0, len(arr) - 1)


data = pandas.read_csv("data.csv")
data1 = list(data["numbers"])
data2 = list(data["numbers"])

start = time.perf_counter()
insertion_sort(data1)
end = time.perf_counter()
insertion_time = end - start
print(data1)
start = time.perf_counter()
mergesort(data2)
end = time.perf_counter()
merge_time = end - start
print(data2)
print("The time taken by insertion sort to sort 500 numbers is : ",
insertion_time)
print("The time taken by merge sort to sort 500 numbers is : ", merge_time)
print(
    "the minimum number in data = ",
    min(data1),
    "the maximum number in data =",
    max(data1),
)


time_data = pandas.DataFrame({
   "insertion sort": insertion_time,
    "merge sort" : merge_time
    },
      index=[]
)
time_data.plot(kind="bar", color = ["red","green"])
plt.xlabel("Execution Time")
plt.ylabel("Time (seconds)")
plt.title("Comparison of Insertion Sort and Merge Sort")
plt.legend(title="Sorting Algorithm")
plt.show()



# Q2 =>
def put_john_age_mid(arr, john_age):
    arr.append(john_age)
    mergesort(arr)
    john_post = arr.index(john_age)
    left_arr = arr[0:john_post]
    right_arr = arr[(john_post + 1) : len(arr)]

    left_length = len(left_arr)
    right_length = len(right_arr)
```

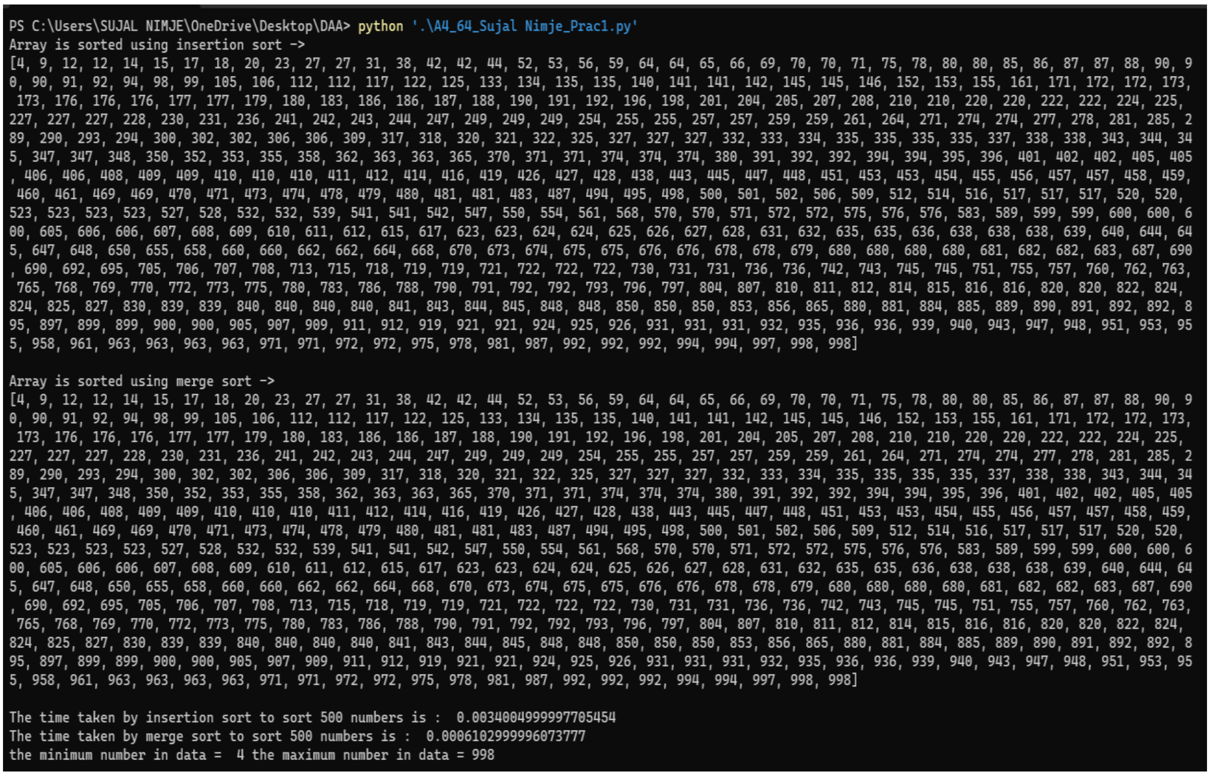


```
    if left_length == right_length:
        return arr
    elif left_length > right_length:
        make_left_equal(left_arr, right_arr, left_length, right_length,
john_age)
    else:
        make_right_equal(left_arr, right_arr, left_length, right_length,
john_age)

    arr = left_arr
    arr.append(john_age)
    arr.extend(right_arr)
    return arr


def make_left_equal(left_arr, right_arr, left_length, right_length, john_age):
    choice = int(
        input(
            "would you like to add a student or delete write 1 for add and 0
for delete : "
        )
    )

    if choice == 0:
        print(
            "we discarded",
            left_length - right_length,
            "students to place john at middle",
        )
        while left_length > right_length:
            left_arr.pop(0)
            left_length = len(left_arr)

    else:
        print(
            "we invited", left_length - right_length, "students to place john
at middle"
        )
        while left_length > right_length:
            if len(right_arr) == 0:
                right_arr.append(john_age + 1)
                right_length += 1
                continue
            right_arr.append(right_arr[right_length - 1] + 1)
            right_length += 1


def make_right_equal(left_arr, right_arr, left_length, right_length,
john_age):
    choice = int(
        input(
            "would you like to add a student or delete write 1 for add and 0
for delete : "
```

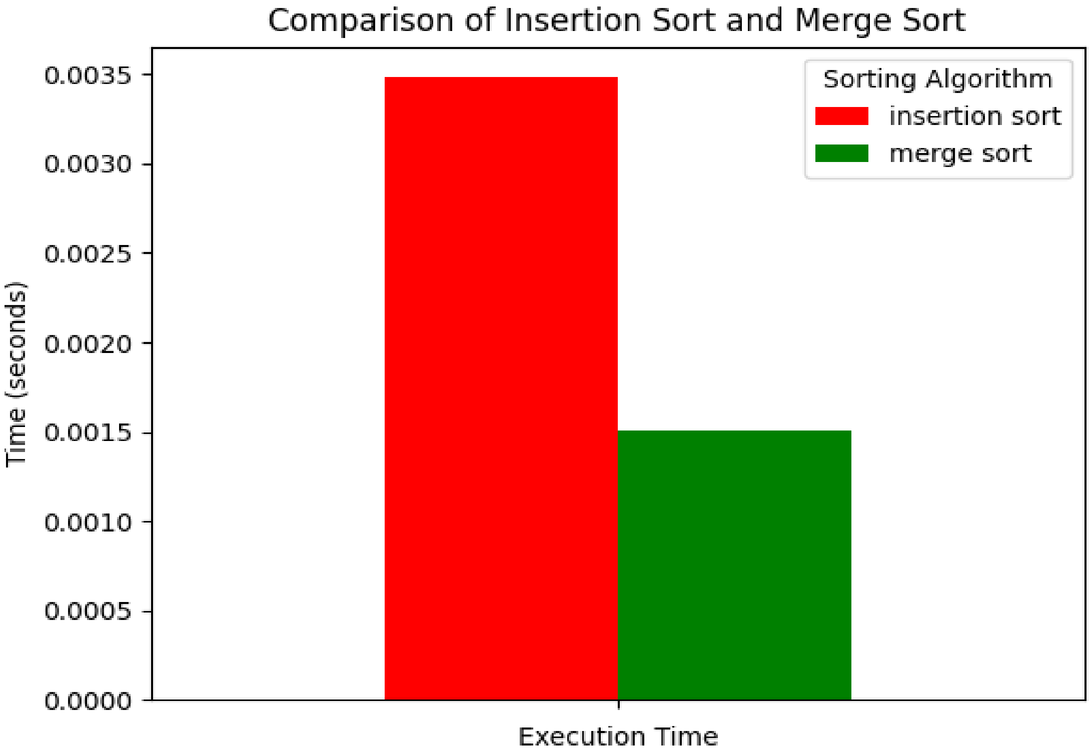


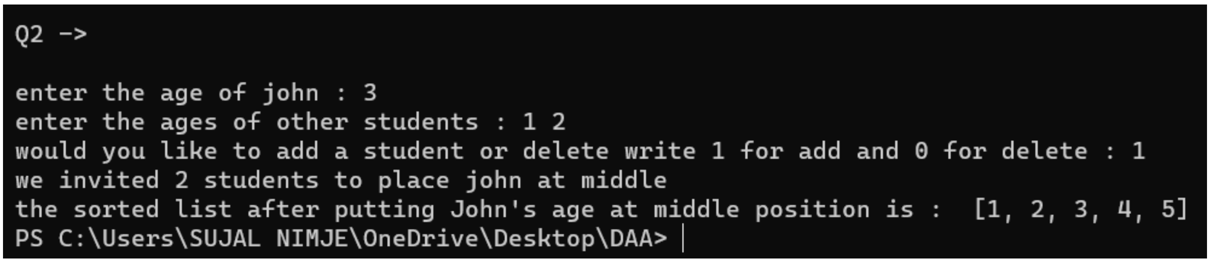
```python
        )
    )
    if choice == 0:
        print(
            "we discarded",
            right_length - left_length,
            "students to place john at middle",
        )
        while right_length > left_length:
            right_arr.pop()
            right_length -= 1

    else:
        print(
            "we invited", right_length - left_length, "students to place john at middle"
        )

        while right_length > left_length:
            if len(left_arr) == 0:
                if john_age <= 1:
                    left_arr.append(john_age)
                else:
                    left_arr.append(john_age - 1)

                left_length += 1
                continue

            if left_arr[0] <= 1:
                append_value = 0
            else:
                append_value = 1
            left_arr.append(left_arr[0] - append_value)
            left_length += 1


john_age = int(input("enter the age of john : "))

age_list = list(input("enter the ages of other students : ").split(" "))
age_list = [int(x) for x in age_list]
age_list = put_john_age_mid(age_list, john_age)
print("the sorted list after putting John's age at middle position is : ", age_list)
```

Output :



```
PS C:\Users\SUJAL NIMJE\OneDrive\Desktop\DAA> python '.\A4_64_Sujal Nimje_Prac1.py'
Array is sorted using insertion sort ->
[4, 9, 12, 12, 14, 15, 17, 18, 20, 23, 27, 27, 31, 38, 42, 42, 44, 52, 53, 56, 59, 64, 64, 65, 66, 69, 70, 70, 71, 75, 78, 80, 80, 85, 86, 87, 87, 88, 90, 9
0, 90, 91, 92, 94, 98, 99, 105, 106, 112, 112, 117, 122, 125, 133, 134, 135, 135, 140, 141, 141, 142, 145, 145, 146, 152, 153, 155, 161, 171, 172, 172, 173,
 173, 176, 176, 176, 177, 177, 179, 180, 183, 186, 186, 187, 188, 190, 191, 192, 196, 198, 201, 204, 205, 207, 208, 210, 210, 220, 220, 222, 222, 224, 225,
227, 227, 227, 228, 230, 231, 236, 241, 242, 243, 244, 247, 249, 249, 249, 254, 255, 255, 257, 257, 259, 259, 261, 264, 271, 274, 274, 277, 278, 281, 285, 2
89, 290, 293, 294, 300, 302, 302, 306, 306, 309, 317, 318, 320, 321, 322, 325, 327, 327, 327, 332, 333, 334, 335, 335, 335, 335, 337, 338, 338, 343, 344, 34
5, 347, 347, 348, 350, 352, 353, 355, 358, 362, 363, 363, 365, 370, 371, 371, 374, 374, 374, 380, 391, 392, 392, 394, 394, 395, 396, 401, 402, 402, 405, 405
, 406, 406, 408, 409, 409, 410, 410, 410, 411, 412, 414, 416, 419, 426, 427, 428, 438, 443, 445, 447, 448, 451, 453, 453, 454, 455, 456, 457, 457, 458, 459,
 460, 461, 469, 469, 470, 471, 473, 474, 478, 479, 480, 481, 481, 483, 487, 494, 495, 498, 500, 501, 502, 506, 509, 512, 514, 516, 517, 517, 517, 520, 520,
523, 523, 523, 523, 527, 528, 532, 532, 539, 541, 541, 542, 547, 550, 554, 561, 568, 570, 570, 571, 572, 572, 575, 576, 576, 583, 589, 599, 599, 600, 600, 6
00, 605, 606, 606, 607, 608, 609, 610, 611, 612, 615, 617, 623, 623, 624, 624, 625, 626, 627, 628, 631, 632, 635, 635, 636, 638, 638, 638, 639, 640, 644, 64
5, 647, 648, 650, 655, 658, 660, 660, 662, 662, 664, 668, 670, 673, 674, 675, 675, 676, 676, 678, 678, 679, 680, 680, 680, 680, 681, 682, 682, 683, 687, 690
, 690, 692, 695, 705, 706, 707, 708, 713, 715, 718, 719, 719, 721, 722, 722, 722, 730, 731, 731, 736, 736, 742, 743, 745, 745, 751, 755, 757, 760, 762, 763,
 765, 768, 769, 770, 772, 773, 775, 780, 783, 786, 788, 790, 791, 792, 792, 793, 796, 797, 804, 807, 810, 811, 812, 814, 815, 816, 816, 820, 820, 822, 824,
824, 825, 827, 830, 839, 839, 840, 840, 840, 840, 841, 843, 844, 845, 848, 848, 850, 850, 850, 853, 856, 865, 880, 881, 884, 885, 889, 890, 891, 892, 892, 8
95, 897, 899, 899, 900, 900, 905, 907, 909, 911, 912, 919, 921, 921, 924, 925, 926, 931, 931, 931, 932, 935, 936, 936, 939, 940, 943, 947, 948, 951, 953, 95
5, 958, 961, 963, 963, 963, 963, 971, 971, 972, 972, 975, 978, 981, 987, 992, 992, 992, 994, 994, 997, 998, 998]

Array is sorted using merge sort ->
[4, 9, 12, 12, 14, 15, 17, 18, 20, 23, 27, 27, 31, 38, 42, 42, 44, 52, 53, 56, 59, 64, 64, 65, 66, 69, 70, 70, 71, 75, 78, 80, 80, 85, 86, 87, 87, 88, 90, 9
0, 90, 91, 92, 94, 98, 99, 105, 106, 112, 112, 117, 122, 125, 133, 134, 135, 135, 140, 141, 141, 142, 145, 145, 146, 152, 153, 155, 161, 171, 172, 172, 173,
 173, 176, 176, 176, 177, 177, 179, 180, 183, 186, 186, 187, 188, 190, 191, 192, 196, 198, 201, 204, 205, 207, 208, 210, 210, 220, 220, 222, 222, 224, 225,
227, 227, 227, 228, 230, 231, 236, 241, 242, 243, 244, 247, 249, 249, 249, 254, 255, 255, 257, 257, 259, 259, 261, 264, 271, 274, 274, 277, 278, 281, 285, 2
89, 290, 293, 294, 300, 302, 302, 306, 306, 309, 317, 318, 320, 321, 322, 325, 327, 327, 327, 332, 333, 334, 335, 335, 335, 335, 337, 338, 338, 343, 344, 34
5, 347, 347, 348, 350, 352, 353, 355, 358, 362, 363, 363, 365, 370, 371, 371, 374, 374, 374, 380, 391, 392, 392, 394, 394, 395, 396, 401, 402, 402, 405, 405
, 406, 406, 408, 409, 409, 410, 410, 410, 411, 412, 414, 416, 419, 426, 427, 428, 438, 443, 445, 447, 448, 451, 453, 453, 454, 455, 456, 457, 457, 458, 459,
 460, 461, 469, 469, 470, 471, 473, 474, 478, 479, 480, 481, 481, 483, 487, 494, 495, 498, 500, 501, 502, 506, 509, 512, 514, 516, 517, 517, 517, 520, 520,
523, 523, 523, 523, 527, 528, 532, 532, 539, 541, 541, 542, 547, 550, 554, 561, 568, 570, 570, 571, 572, 572, 575, 576, 576, 583, 589, 599, 599, 600, 600, 6
00, 605, 606, 606, 607, 608, 609, 610, 611, 612, 615, 617, 623, 623, 624, 624, 625, 626, 627, 628, 631, 632, 635, 635, 636, 638, 638, 638, 639, 640, 644, 64
5, 647, 648, 650, 655, 658, 660, 660, 662, 662, 664, 668, 670, 673, 674, 675, 675, 676, 676, 678, 678, 679, 680, 680, 680, 680, 681, 682, 682, 683, 687, 690
, 690, 692, 695, 705, 706, 707, 708, 713, 715, 718, 719, 719, 721, 722, 722, 722, 730, 731, 731, 736, 736, 742, 743, 745, 745, 751, 755, 757, 760, 762, 763,
 765, 768, 769, 770, 772, 773, 775, 780, 783, 786, 788, 790, 791, 792, 792, 793, 796, 797, 804, 807, 810, 811, 812, 814, 815, 816, 816, 820, 820, 822, 824,
824, 825, 827, 830, 839, 839, 840, 840, 840, 840, 841, 843, 844, 845, 848, 848, 850, 850, 850, 853, 856, 865, 880, 881, 884, 885, 889, 890, 891, 892, 892, 8
95, 897, 899, 899, 900, 900, 905, 907, 909, 911, 912, 919, 921, 921, 924, 925, 926, 931, 931, 931, 932, 935, 936, 936, 939, 940, 943, 947, 948, 951, 953, 95
5, 958, 961, 963, 963, 963, 963, 971, 971, 972, 972, 975, 978, 981, 987, 992, 992, 992, 994, 994, 997, 998, 998]

The time taken by insertion sort to sort 500 numbers is :  0.003400499999705454
The time taken by merge sort to sort 500 numbers is :  0.0006102999996073777
the minimum number in data =  4 the maximum number in data = 998
```





```
Q2 ->

enter the age of john : 3
enter the ages of other students : 1 2
would you like to add a student or delete write 1 for add and 0 for delete : 1
we invited 2 students to place john at middle
the sorted list after putting John's age at middle position is :  [1, 2, 3, 4, 5]
PS C:\Users\SUJAL NIMJE\OneDrive\Desktop\DAA> 
```