Candidate name: ....Chan Zun Mun Terence........................................

Centre number: ...3011...........................................

Index number: ..................................................

Programming language used: ....Python 3.............................................

|  |
| --- |
| **Question 1** |
| **Evidence 1**  *Program code*  data = []  def readData(dataFile):  global data  dataFile.seek(0)  data = dataFile.read().split()  for i in range(len(data)):  data[i] = int(data[i])  print("Data Read!")  print(data)  def menu():  dataFile = open("ADMISSIONS-DATA.TXT")  userin = "" #User Input  while userin != "4":  print("""Menu  1. Read F=file Data  2. Bubble Sort  3. Insertion Sort""")  userin = input("Option: ")  if userin == "1":  readData(dataFile)  elif userin == "2":  newdata = list(tuple(data))  bubblesort(newdata)  print(newdata)  elif userin == "3":  newdata = list(tuple(data))  insertionSort(newdata)  print(newdata)  dataFile.close()  menu() |
| **Evidence 2**  *Program code*  def bubblesort(Admissions):  comparisons = 0  UpperBound = len(Admissions)  NoSwaps = False  while NoSwaps == False:  NoSwaps = True  for Posn in range(0, UpperBound-1):  comparisons += 1  if Admissions[Posn] > Admissions[Posn + 1]:  # swap  NoSwaps = False  Temp = Admissions[Posn]  Admissions[Posn] = Admissions[Posn + 1]  Admissions[Posn + 1] = Temp  print(f"Bubble Sort Comparisons Made: {comparisons}") |
| **Evidence 3**  The sort algorithm used was :…Insertion Sort…………………………  *Program code*  def insertionSort(lis):  comparisons = 0  for i in range(1, len(lis)):  pivot = lis[i]  j = i-1  while lis[j] > pivot and j >= 0:  comparisons += 1  #Shift Rightwards  lis[j+1] = lis[j]  j -= 1  comparisons += 1 #Comparison where it exits loop  lis[j+1] = pivot  print(f"Insertion Sort Comparisons Made: {comparisons}") |
| **Evidence 4**  *Screenshot* |
| **Evidence 5**  *Screenshot* |

2015 P2 Q5

a)

InsertionSortTester

Perform Insertion

Sort

Output Sorted Values

In the Array

Validation

Output Initial

Values of Array

Input Integer Values

into the Array

b)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Values | | | |  | | |
| [1] | [2] | [3] | [4] | i | j | Temp |
| 6 | 8 | 2 | 1 | 2 | 1 | 8 |
|  | 8 |  |  |  |  |  |
|  |  | 8 |  | 3 | 2 | 2 |
|  | 6 |  |  |  | 1 |  |
| 2 |  |  |  |  | 0 |  |
|  |  |  | 8 | 4 | 3 | 1 |
|  |  | 6 |  |  | 2 |  |
|  | 2 |  |  |  | 1 |  |
| 1 |  |  |  |  | 0 |  |

c) For an array that is sorted, the algorithm would still have to overwrite each value in the array, which would take up more time and be inefficient.

d) One test case would be to have a list sorted in reverse, to test in the worst case scenario, if the program can correctly compare the elements in the array and shift and sort the list. Another test case would be to have a fully sorted list, to test the best case scenario, if the program can compare the elements correctly and know not to modify the list