Screenshots or Program Listings must be copied into appropriate cells in the following table.

Examiners must be able to read the contents including any screenshots without the use of a magnifying glass.

Answers that are not readable will not be awarded any marks.

Save this evidence document at regular intervals, for example every 5 minutes.

|  |
| --- |
| **Question 1** |
| **Part 1(a)** |
| *TheData = [20, 3, 4, 8, 12, 99, 4, 26, 4]* |
| **Part 1(b)** |
| *def InsertionSort(TheData):*      for count in range(len(*TheData*)):          DataToInsert = *TheData*[count]          inserted = 0          nextvalue = count - 1          while (nextvalue >= 0 and inserted != 1):              if DataToInsert < *TheData*[nextvalue]:  *TheData*[nextvalue + 1] = *TheData*[nextvalue]                  nextvalue = nextvalue - 1  *TheData*[nextvalue + 1] = DataToInsert              else:                  inserted = 1 |
| **Part 1(c)** |
| def printdata(TheData):  for number in TheData:  print(number) |
| **Part 1(d)(i)** |
| print("Before sorting: ")  printdata(TheData)  InsertionSort(TheData)  print("After sorting: ")  printdata(TheData) |
| **Part 1(d)(ii)** |
|  |
| **Part 1(e)(i)** |
| def finddata(TheData):  searchnum = int(input("Input the number you want to find: "))  for data in TheData:  if searchnum == data:  print("found")  return True  print("not found")  return False |
| **Part 1(e)(ii)** |
|  |
| **Question 2** |
| **Part 2(a)** |
| *class HiddenBox():*  *# BoxName String*  *# Creator String*  *# DateHidden Datetime*  *# GameLocation String*  *# LastFinds Array[10][2] of String*  *# Active Boolean* |
| **Part 2(b)** |
| *def \_\_init\_\_(self, BoxName, Creator, DateHidden, GameLocation):*  *self.BoxName = BoxName*  *self.Creator = Creator*  *self.DateHidden = DateHidden*  *self.GameLocation = GameLocation*  *self.LastFinds = [["" for i in range(2)] for j in range(10)]*  *self.Active = False* |
| **Part 2(c)** |
| *def GetBoxName(self):*  *return self.BoxName*  *def GetGameLocation(self):*  *return self.GameLocation* |
| **Part 2(d)(i)** |
| *TheBoxes = [HiddenBox("", "", "", "") for i in range(10000)]* |
| **Part 2(d)(ii)** |
| *def NewBox(TheBoxes):*  *BoxName = input("Enter the name of the box: ")*  *Creator = input("Enter the creator's name: ")*  *DateHidden = input("Enter the date the box was hidden: ")*  *Location = input("Enter the location of the box: ")*  *TheBoxes.append(HiddenBox(BoxName, Creator, DateHidden, Location))* |
| **Part 2(d)(iii)** |
| *NewBox(TheBoxes)* |
| **Part 2(e)** |
| *class PuzzleBox(HiddenBox):*  *def \_\_init\_\_(self, BoxName, Creator, DateHidden, GameLocation, PuzzleText, Solution):*  *super().\_\_init\_\_(BoxName, Creator, DateHidden, GameLocation)*  *self.PuzzleText = PuzzleText #String*  *self.Solution = Solution #String* |
| **Question 3** |
| **Part 3(a)** |
| *QueueData = ["" for i in range(20)]*  *startpointer = 0*  *endpointer = 0* |
| **Part 3(b)** |
| *def Enqueue(QueueData, NewData):*  *global endpointer*  *if endpointer != 20:*  *QueueData[endpointer] = NewData*  *endpointer += 1*  *return True*  *return False* |
| **Part 3(c)** |
| *def ReadFile():*  *filename = input("Input the text file name: ")*  *try:*  *f = open(filename, "r")*  *f.close()*  *except:*  *print("File doesn't exist")*  *return -1*  *with open(filename, "r") as f:*  *notfull = True*  *NewData = f.readline()*  *while notfull == True and NewData != "":*  *notfull = Enqueue(QueueData, NewData)*  *NewData = f.readline()*  *if notfull:*  *return 1*  *else:*  *return 2* |
| **Part 3(d)(i)** |
| *returnvalue = ReadFile()*  *if returnvalue == 1:*  *print("All items added")*  *elif returnvalue == 2:*  *print("Queue is full")*  *else:*  *print("File does not exist")* |
| **Part 3(d)(ii)** |
|  |
| **Part 3(e)** |
| *def Remove(QueueData):*  *global startpointer*  *if QueueData[startpointer] != "" and QueueData[startpointer+1] != "":*  *removedstring = QueueData[startpointer] + " " + QueueData[startpointer+1]*  *QueueData[startpointer] = ""*  *QueueData[startpointer+1] = ""*  *startpointer += 2*  *return removedstring*  *else:*  *return "No Items"* |