Marking Scheme

1	In cell **C3** enter an appropriate formula to extract the year of purchase of the car from the purchase date given, and use it to complete the **Year of Purchase** column. =RIGHT(B3,4) - 1 mark
2	Use an appropriate function to search for the **Annual Interest Rate (%)** and use it to complete the **Annual Interest Rate (%)** column. =VLOOKUP(E3,$A$14:$B$17,2,FALSE) - 1 mark - 1 mark for $ sign and correct column values
3	In cell **H3** enter an appropriate formula to calculate the monthly principal payment of the car, and use it to complete the **Monthly Principal Payment ($)** column. =PPMT(G3/100/12,F3,10*12,D3) - 1 mark - 1 mark for correct column values
4	In cell **I3** enter an appropriate formula to calculate the monthly interest payment of the car, and use it to complete the **Monthly Interest Payment ($)** column. =IPMT(G3/100/12,F3,E3*12,D3) - 1 mark - 1 mark for correct column values
5	In cell **J3** enter an appropriate formula to calculate the total monthly payment of the car, and use it to complete the **Total Monthly Payment ($)** column. =H3+I3 - 1 mark
6	The company uses a credit status system to classify its customer. There are 3 groups: A: total monthly payment is less than $1000 B: total monthly payment is less than $2000 C: total monthly payment is $2000 or more Use a conditional statement in cell **K3** to indicate if the customer belongs to group **A**, **B** or **C**, and use it to complete the **Credit Status** column. =IF(J3>-1000,"A",IF(J3>-2000,"B","C")) - 1 mark for outer condition, 1 mark for inner condition

7

```
word = input ("Enter a word: ")
longest_word = word [Q7(b): M1]
shortest_word = word
num_words = 9  [Q7(a): A1]
palindrome = 0 [Q7(c): M1]

for count in range (num_words):
    word = input ("Enter a word: ")
    if len (longest_word) < len(word):
        longest_word = word [Q7(b): M1]
    if len(shortest_word) > len(word):
        shortest_word = word
    if word == word [::-1]: [Q7(c): M1]
        palindrome += 1     [Q7(c): M1]

print ("Longest word is " + longest_word) [Q7(b): A1]
print ("Shortest word is " + shortest_word)
print ("There are " + str(palindrome) + " palindrome word(s)")
[Q7(c): A1]
```

8

```
word = input ("Enter a word: ")
longest_word = word
shortest_word = word
num_words = int(input("Enter the number of words to input"))
[Q8: 2 marks]
palindrome = 0
for count in range (num_words):
    word = input ("Enter a word: ")
    if len (longest_word) < len(word):
        longest_word = word
    if len(shortest_word) > len(word):
        shortest_word = word
    if word == word [::-1]:
        palindrome += 1

print ("Longest word is " + longest_word)
print ("Shortest word is " + shortest_word)
print ("There are " + str(palindrome) + " palindrome word(s)")
```

9

```
vending_list= ["Coffee", 23, "Coke", 12,
```

```
                "Tea", 19, "Milo", 18] #do not modify this line

total_order = [0,0,0,0]
message = "Please enter your choice (1: Coffee, 2: Coke, " \
"3: Tea, 4: Milo, 0: Confirm Selection): "

order = int(input(message))
if order != 0:
    qty = int(input("Enter the quantity to dispense: ")) #1m

while order !=0:
    if order == 1: #1m
        if vending_list[1] - qty >= 0:
            total_order [0] = qty
        else:
            print ("Quantity unavailable, " \
                   + str(vending_list[1]) + " remaining")

    elif order == 2:
        if vending_list[3] - qty >= 0:
            total_order [1] = qty #1m
        else:
            print ("Quantity unavailable, " \
                   + str(vending_list[3]) + " remaining")

    elif order == 3:
        if vending_list[5] - qty >= 0: #1m
            total_order [2] = qty
        else:
            print ("Quantity unavailable, " \
                   + str(vending_list[5]) + " remaining") #1m

    elif order == 4: #1m
        if vending_list[7] - qty >= 0:
            total_order [3] = qty
        else:
            print ("Quantity unavailable, " \
                   + str(vending_list[7]) + " remaining")
    else:
        print ("Input error.")


    order = int(input(message))
    if order != 0:
        qty = int(input("Enter the quantity to dispense: "))

vending_list[1] = vending_list[1] - total_order[0] #1m
vending_list[3] = vending_list[3] - total_order[1]
vending_list[5] = vending_list[5] - total_order[2]
vending_list[7] = vending_list[7] - total_order[3] #1m
```

```
print ("You have bought the following items: ")

i = 0 #1m
while i in range (len(total_order)):
    if total_order[i] > 0:
        print (vending_list[i*2] + ": " + str(total_order[i])) #1m
    i += 1
```

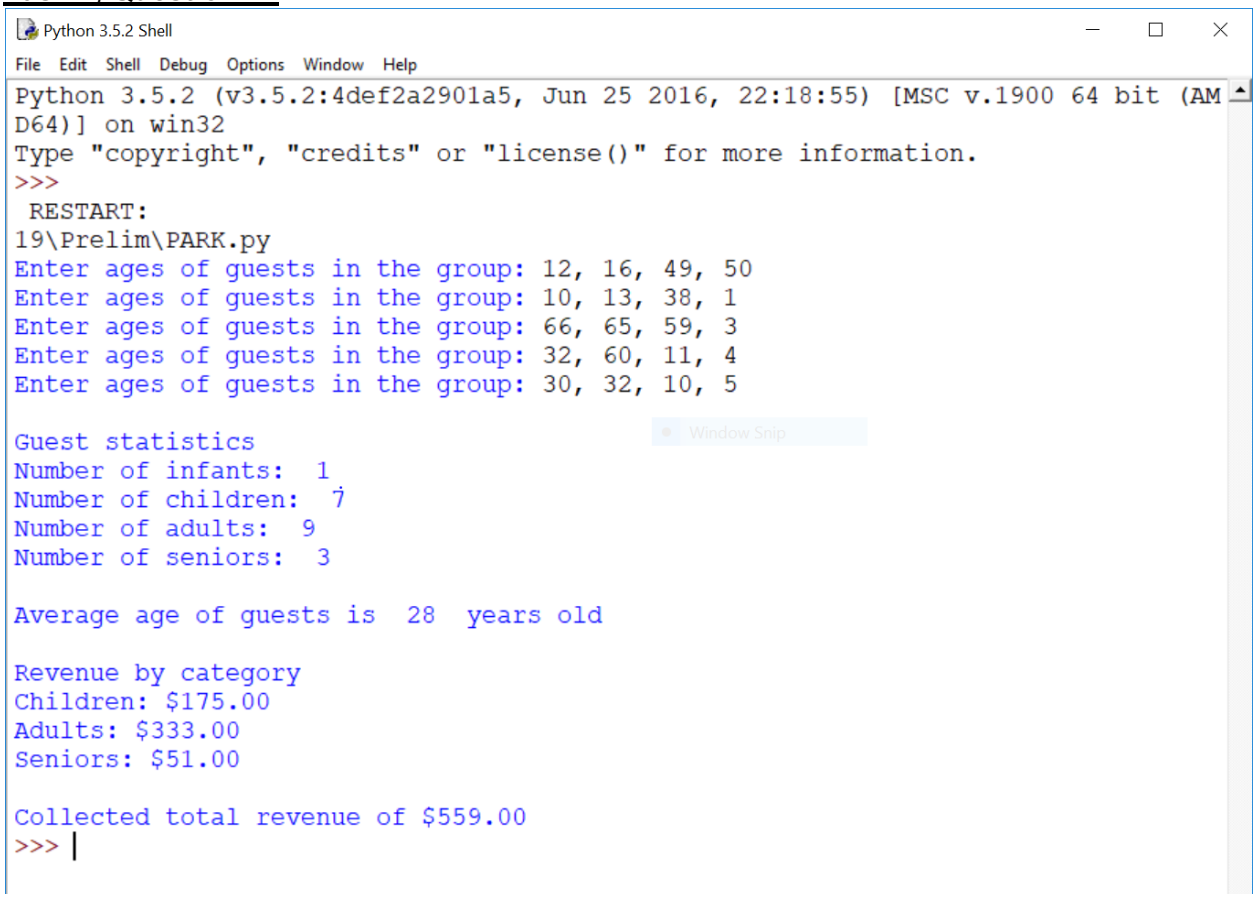
Task 4, Question 10

```
sum_age = 0
revenue = 0
num_guest = 0
guest_lst = [0, 0, 0, 0] #Num of infant, children, adult, senior
price_lst = [0, 56, 76, 38] #Price for infant, children, adult, senior

def is_valid(string):
  age_lst = string.split(", ")
  for age in age_lst:
    if not age.isdigit():
      return False
    elif int(age) < 0 or int(age) > 120:
      return False
  return True

for group_num in range(5):
  group = input("Enter ages of guests in the group: ")
  is_invalid = not is_valid(group)
  while is_invalid:
    group = input("Enter valid ages of guests: ")
    is_invalid = not is_valid(group)
  age_lst = group.split(", ")
  for age in age_lst:
    age = int(age)
    sum_age += age
    num_guest += 1
    if age < 3:
      guest_lst[0] += 1
    elif age < 13:
      guest_lst[1] += 1
    elif age < 60:
      guest_lst[2] += 1
    else:
      guest_lst[3] += 1

print("")
print("Guest statistics")
print("Number of infants: ", guest_lst[0])
print("Number of children: ", guest_lst[1])
print("Number of adults: ", guest_lst[2])
print("Number of seniors: ", guest_lst[3])
print("")
```

```
print("Average age of guests is ", int(sum_age/num_guest), " years old")
print("")
print("Revenue by category")
print("Children: $%.2f" % (guest_lst[1]*price_lst[1]))
print("Adults: $%.2f" % (guest_lst[2]*price_lst[2]))
print("Seniors: $%.2f" % (guest_lst[3]*price_lst[3]))
print("")
print("Collected total revenue of $%.2f" % (guest_lst[1]*price_lst[1] + guest_lst[2]*price_lst[2] +
guest_lst[3]*price_lst[3]))
```

- Initialise counter variables with suitable data types
- Use of for / while loops correctly
- Obtains (5 valid sets of) user input
- Manipulation of input data to retrieve ages (e.g. string slicing / list.split())
- Type forcing of input to integer
- Validates input are integers and ensures valid input
- Validates input are between 0 and 120 (inclusive) and ensures valid input
- Use of conditional statements correctly to check for age group
- Rounds down average age to nearest year (e.g. round() / int() / quotient)
- Prints currency to 2 decimal places (e.g. using print formatting / hard code ".00")

Task 4, Question 11



- Guest statistics block printed accurately & correctly

	• Average age sentence printed accurately & correctly (excludes rounding inaccuracies) • Revenue by category block printed accurately & correctly • Total revenue block printed accurately & correctly (excludes d.p. inaccuracies) • An empty line between each block (total of 3 empty lines) **<u>Task 4, Question 12</u>** Same code as Question 11 • Amend for / while loop iterations correctly to cater for varying lengths of input • Validates that there must be at least 1 guest entered each time • Prints output accurately and correctly **<u>Task 4, Question 13</u>** • Asks user for input on number of entries before initiating the loop / Checks for a specific end-loop process (e.g. blank entry / typing “END”) Prints private test case accurately and correctly (1st input: “END”)