COMPUTING (7155/02)
Marking Scheme

Task 1

Qn	Answer	Marks
1	**one** mark for correct top formula, **one** mark for column =MID(B7,4,3)	2
2	**one** mark for working vlookup formula **one** mark for multiplying Quantity Ordered column **one** mark for column =VLOOKUP(C7,$B$41:$D$46,3,FALSE)*D7 Or =VLOOKUP(C7,$B$42:$D$46,3,FALSE)*D7	3
3	**one** mark for correct top formula, **one** mark for column =MID(B7, 12, 2) OR =RIGHT(B7,2)	1
4	**one** mark for correct top formula, **one** mark for column =IF(OR(F7="SG", F7="MY"), "Discount", "Surcharge")	2
5	**one** mark for correct top formula, **one** mark for column =IF(G7="Discount", 0.88*E7, 1.2*E7)	2
	Total	10

Task 2

Qn		Answer	Marks
6	(a)	`while counter < 10:`	
	(b)	`total = 0`	
		`total += age` `average = total // counter`	
		`print("The average age is ", average)`	
	(c)	`while age < 21 or age > 65:`	
		`age = int(input("Enter an age between 21 and 65: "))`	
7		`num_emp = int(input("Enter total number of employees: "))`	
		`while counter < num_emp:`	
		Total	

Task 3

Qn	Answer	Marks

8	Single errors are underlined
	
	`result = []`
	
	`print("***********The search for ARMSTRONG numbers!***********")`
	`limit = int(input("Range is 1 to n inclusive. \nState your n: "))`
	`print("Checking for ARMSTRONG numbers from 1 to {}".format(limit))`
	
	`for number in range(1, limit+1)`
	`    ##sum_digits = 0`
	`    power = len(number)`
	`    check = number`
	`    remainder = 0`
	
	`    while check != 0:`
	`        remainder = number%10`
	`        sum_digits += remainder*power`
	`        check = check%10`
	
	`    if sum_digits > number:`
	`    result.append(number)`
	
	`    print("The ARMSTRONG numbers are {}: ".format(result))`
	
	`##List of ARMSTRONG numbers`
	`##1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407,`
	`##1634, 8208, 9474, 54748, 92727, 93084, 548834, ...`	

	Corrected lines.	
	`limit = int(input("Range is 1 to n inclusive. \nState your n: "))`	1
	`for number in range(1, limit+1):`	1
	`sum_digits = 0` **(added)**	1
	`power = len(str(number))`	1
	`remainder = check%10`	1
	`sum_digits += remainder**power`	1
	`check = check//10`	1
	`if sum_digits == number:`	1
	(indented) `result.append(number)`	1
	`print("The ARMSTRONG numbers are {}: ".format(result))` **(remove indent)**	1
	Total	10

Suggested Solution:

```
alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

choice = int(input("Enter 1 for Single-level encryption \n   or 2
for Double-level encryption: "))

plaintext = input("Please enter your plaintext: ")
plaintext = plaintext.upper()

encrypted = ""
encrypted_2 = ""

for c in plaintext:
    if c in alpha:
        encrypted += alpha[25-(alpha.find(c))]
    else:
        encrypted += c

if choice == 2:
    encrypted = encrypted[::-1]
    for i in range(0, len(encrypted), 2):
        encrypted_2 += encrypted[i:i+2]
        encrypted_2 += "%"
    encrypted = encrypted_2

print("Encrypted message:", encrypted)
```

	Total	**20**