MARKING SCHEME

Question	Answer	Marks
	Task 1	
1	=LEFT(A4,1)	1
2	=HLOOKUP(D4,$K$5:$O$6,2,TRUE) (1 mark for HLookup, 1 mark for correct arguments)	2
3	=F4*E4	1
4	=IF(G4>75,"YES","NO") (1 mark for using IF, 1 mark for correct condition	2
5	1 mark for condition, 1 mark for formatting	2
6	=COUNTIF(C$4:C$22,"N") (1/2 mark for each correct function)	2
	Task 2	

7	`ID = ''`
	`count_SG = 0`
	
	`for i in range(5):`
	`  ID = input("Enter ID: ")`
	`  if ID[0] == "S" or ID[0] == "T":`
	`  print("Welcome home!")`
	
	`  count_SG += 1`
	`  else:`
	`  print("Welcome to Singapore!")`
	
	`print("Number of Singaporeans returning home: ", count_SG)`	

8	`ID = ''`
	`count_SG = 0`
	`num = int(input("Enter number of entries: "))`
	
	`for i in range(num):`
	`  ID = input("Enter ID: ")`
	`  while len(ID)!= 9:`
	`  ID = input("Incorrect length. Please enter again:")`
	`  if ID[0] == "S" or ID[0] == "T":`
	`  print("Welcome home!")`
	`  count_SG += 1`
	`  else:`
	`  print("Welcome to Singapore!")`
	
	`print("Number of Singaporeans returning home: ", count_SG)`	

Question	Answer	Marks
	Task 3	

9	`string = input("Enter string: ")`
	`index = int(input("Enter the index/indices of characters you wish to extract in a,b,c format: "))`
	`indx_lst == index.split("")`
	
	`new_string = ''`
	`final_string = ''`
	
	`for j in range(string):`
	`  if string[j] == '':`
	`  break`
	`  else:`
	`  new_string = new_string + string[j]`
	
	`for i in range(len(indx_lst)):`
	`  indx_lst[i] = indx_lst[i]`
	
	`  for ele in range(indx_lst):`
	`  final_string = final_string + new_string[ele]`
	
	`print("The original sentence is: , string")`
	`print("The index/indices of characters you wish to extract is/are ", indx_lst)`
	`print("The character(s) extracted is/are ", final_string)`	

	`index = input("Enter the index/indices of characters you wish to extract in a,b,c format: ")`	
	`indx_lst = index.split(",")`	
	`for j in range(len(string)):`	
	`  if string[j] == ' ':`	
	`  continue`	
	`  indx_lst[i] = int(indx_lst[i])`	
	`for ele in indx_lst:`	
	`print("The original sentence is: ", string)`	
	Task 4	
10	Input marks in proper format	
	variables set up and initialised for **grades**	
	variables set up and initialised for **number of distinction, credit, pass and failure**	
	use of loop till entry = x	
	Conversion of raw marks into corresponding grade	
	correct calculation of **number of distinction, credit, pass and failures**	

Question	Answer	Marks
	Correct display with format	
11	Validation of length = 4	
	Validation for marks to be between 0 and 100	
	Re-entries of grades if fail	
12	Screenshot: Correct output	
	File in png or jpg format	
13	Extend: input of distinction mark, credit mark and passing mark	
	Validation of distinction mark higher than the other two marks, and credit marks higher than passing mark	

SOLUTIONS

Task 2

(MYVERIFY.py)

```
ID = ''
count_SG = 0

for i in range(5):
    ID = input("Enter ID: ")
    if ID[0] == "S" or ID[0] == "T":
        print("Welcome home!")
        count_SG += 1
    else:
        print("Welcome to Singapore!")

print("Number of Singaporeans returning home: ", count_SG)
```

(NUMVERIFY.py)

```
ID = ''
count_SG = 0
num = int(input("Enter number of entries: "))

for i in range(num):
    ID = input("Enter ID: ")
    while len(ID)!= 9:
        ID = input("Incorrect length. Please enter again:")
    if ID[0] == "S" or ID[0] == "T":
        print("Welcome home!")
        count_SG += 1
    else:
        print("Welcome to Singapore!")

print("Number of Singaporeans returning home: ", count_SG)
```

Task 3

(LIST.py)

```
string = input("Enter string: ")
index = input("Enter the index/indices of characters you wish to
extract in a,b,c format: ")
indx_lst = index.split(",")

new_string = ''
final_string = ''

for j in range(len(string)):
    if string[j] == ' ':
        continue
    else:
        new_string = new_string + string[j]

for i in range(len(indx_lst)):
              indx_lst[i] = int(indx_lst[i])

for ele in indx_lst:
    final_string = final_string + new_string[ele]

print("The original sentence is: ", string)
print("The index/indices of characters you wish to extract is/are
", indx_lst)
print("The character(s) extracted is/are ", final_string)
```

Task 4

(GRADES.py)

```
grades = []
dist, credit, passes, fail = 0,0,0,0

while True:
    marks = input("Enter the marks for EL, MT, Math and Sci:
").split(' ')
    if marks[0] == 'x':
        break
    else:
        validity = True
        for i in range(len(marks)):
            marks[i] = int(marks[i])
            if marks[i] >= 75:
                grades.append("D")
            elif marks[i] >= 65:
                grades.append("C")
            elif marks[i] >= 50:
                grades.append("P")
            else:
                grades.append("F")

print("\n\nGrades for \t EL \t MT \t Math \t Sci")
for j in range (0,len(grades)-1,4):
    print("\t\t", grades[j],"\t",
grades[j+1],"\t",grades[j+2],"\t",grades[j+3])

for k in range(len(grades)):
    if grades[k] == "D":
        dist +=1
    elif grades[k] == "C":
        credit +=1
    elif grades[k] == "P":
        passes +=1
    else:
        fail +=1

print("\n\nNumber of distinctions: \t", dist)
print("Number of credits: \t\t", credit)
print("Number of passes: \t\t", passes)
print("Number of failures: \t\t", fail)
```

(VERIFYGRD.py)

```
grades = []
dist, credit, passes, fail = 0,0,0,0

while True:
    marks = input("Enter the marks for EL, MT, Math and Sci:
").split(' ')
    if marks[0] == 'x':
        break
    else:
        validity = True
        if len(marks) == 4:
            for i in range(len(marks)):
                marks[i] = int(marks[i])
                if marks[i] > 100 or marks[i] < 0:
                    print("Error, marks should be between 0 and 100
inclusive")
                    validity = False
        else:
            print("Error, there should only be 4 entries")
            validity = False

        if validity == True:
            for j in range(len(marks)):
                if marks[j] >= 75:
                    grades.append("D")
                elif marks[j] >= 65:
                    grades.append("C")
                elif marks[j] >= 50:
                    grades.append("P")
                else:
                    grades.append("F")


print("\n\nGrades for \t EL \t MT \t Math \t Sci")
for j in range (0,len(grades),4):
    print("\t\t", grades[j],"\t",
grades[j+1],"\t",grades[j+2],"\t",grades[j+3])

for k in range(len(grades)):
    if grades[k] == "D":
        dist +=1
    elif grades[k] == "C":
        credit +=1
    elif grades[k] == "P":
        passes +=1
    else:
        fail +=1

print("\n\nNumber of distinctions: \t", dist)
print("Number of credits: \t\t", credit)
print("Number of passes: \t\t", passes)
print("Number of failures: \t\t", fail)
```

(VARYGRD.py)

```
grades = []
dist, credit, passes, fail = 0,0,0,0
all_pass = False

while all_pass == False:
    dist_mark = int(input("Enter distinction mark:"))
    cred_mark = int(input("Enter credit mark:"))
    pass_mark = int(input("Enter pass mark:"))
    if dist_mark > cred_mark and cred_mark > pass_mark:
        all_pass = True
    else:
        print("Error! Distinction mark may be lower than credit
mark or credit mark is lower than pass mark")

while True:
    marks = input("Enter the marks for EL, MT, Math and Sci:
").split(' ')
    if marks[0] == 'x':
        break
    else:
        validity = True
        if len(marks) == 4:
            for i in range(len(marks)):
                marks[i] = int(marks[i])
                if marks[i] > 100 or marks[i] < 0:
                    print("Error, marks should be between 0 and 100
inclusive")
                    validity = False
        else:
            print("Error, there should only be 4 entries")
            validity = False

        if validity == True:
            for j in range(len(marks)):
                if marks[j] >= dist_mark:
                    grades.append("D")
                elif marks[j] >= cred_mark:
                    grades.append("C")
                elif marks[j] >= pass_mark:
                    grades.append("P")
                else:
                    grades.append("F")


print("\n\nGrades for \t EL \t MT \t Math \t Sci")
for j in range (0,len(grades),4):
    print("\t\t", grades[j],"\t",
grades[j+1],"\t",grades[j+2],"\t",grades[j+3])

for k in range(len(grades)):
    if grades[k] == "D":
        dist +=1
```

```
    elif grades[k] == "C":
        credit +=1
    elif grades[k] == "P":
        passes +=1
    else:
        fail +=1

print("\n\nNumber of distinctions: \t", dist)
print("Number of credits: \t\t", credit)
print("Number of passes: \t\t", passes)
print("Number of failures: \t\t", fail)
```