COMP3 Recursion Homework

There are two questions in this homework, covering the following three areas:

- Algorithms
- Databases

Make sure you have read the relevant pages in the textbook before attempting these questions.

Please refer to the **homework policy** for guidance of how to complete and submit this homework.

Question 1

A *recursively-defined* function A that takes two integers as parameters is defined below.

```
def A(number, entry):
    if number != items[entry]:
        A(Number, entry+1)
    else:
        print(entry)
```

The following table shows the list named items:

index	value
1	4
2	5
3	8
4	11
5	15
6	19
7	21
8	28
9	33

1. What is meant by a recursively-defined function? (**1 mark**)

   A function that has the ability to call itself.

2. What is the role of the stack when a recursively-defined function is executed? (**1 mark**)

   A stack is used to hold the most recent called values in the recursive function so the next value is known.

3. Dry run the function call A(11,1) using the data provided in the list items

given above. Complete the trace table given below: (**4 marks**)

|Number|Entry|Output|

|:----:|:---:|:----:|

|11|1| |

|11|2| |

|11|3| |

|11|4|4|

| | | |

4. What is the purpose of this algorithm? (**1 mark**)

The purpose may be to find an item inside a list.

1. Give a situation where this algorithm will fail. (**1 mark**)

If a number higher than 33 is input the program will display an error as the list has only 9 items.

1. Suggest a modification to the algorithm that will prevent it from failing. (**1 mark**)

One modification could be:

```
def A(number, entry):
if number != items[entry]:
   if entry == 9:
        print("max items reached")
   else:
      A(Number, entry+1)
else:
   print(entry)
```

1. With an ordered list, of many more entries, what more efficient algorithm could be used to achieve your expressed purpose in part (d)? (**1 mark**)

```
 def A(number, entry):
 if number != items[entry]:
    if entry == len(items):
         print("max items reached")
    else:
       A(Number, entry+1)
 else:
    print(entry)
```

Question 2

Customers placing orders with ABC Ltd for ABC's products have their orders

recorded by ABC in a database. The data requirements for the database system are defined as follows:

- Each product is assigned a unique product code, ProductId and has a product description.
- The quantity in stock of a particular product is recorded.
- Each customer is assigned a unique customer code, CustomerId and has their name, address and telephone number recorded.
- An order placed by a customer will be for one or more products.
- ABC Ltd assigns a unique code to each customer order, ABCOrderNo.
- A customer placing an order must supply a code, CustomerorderNo, which the customer uses to identify the particular order.
- A customer may place one or more orders.
- Each new order from a particular customer will have a different customer order code but two different customers may use, independently, the same values of customer order code.
- Whether an order has been despatched or not will be recorded.
- A particular order will contain one or more lines.
- Each line is numbered, the first is one, the second is two, and so on.
- Each line will reference a specific product and specify the quantity ordered.
- A specific product reference will appear only once in any particular order placed with ABC Ltd.

After **normalisation** the database contains four tables based on the entities:

Customer, **Product**, **Order**, **OrderLine**

1. Below is a partially complete entity-relationship diagram. Show the degree of three more relationships which exist between the given entities. (**3 marks**)

   □

   Answer: http://i.imgur.com/4Skcw5V.png

2. Using the following format:

   TableName(<u>Primary Key</u>, Attribute1, Attribute2, etc.)

   Describe the tables, stating all attributes, for the following entities underlining the primary key in each case.

   1. Product (**2 marks**)

2. Customer (**2 marks**)
3. Order (**3 marks**)
4. OrderLine (**4 marks**)

1. Using the SQL commands select, from, where, order by, write an **SQL statement** to query the database tables for all customer names where the orders have been dispatched. The result of the query is to be ordered in ascending order of ABCOrderNo. (**6 marks**)