Computing GCSE Coursework

Thomas Bass
Candidate 4869
Centre 52423
OCR A453 Programming Project

Made with LaTeX

2016-2017

Summary

1 Objectives **3**
1.1 Task 1 . . . 3
1.2 Task 2 . . . 3
1.3 Task 3 . . . 3

2 Test Plan **4**
2.1 Task 1 . . . 4
2.2 Task 2 . . . 4
2.3 Task 3 . . . 5

3 Pseudocode **6**
3.1 Task 1 . . . 6
3.2 Task 2 . . . 7
3.3 Task 3 . . . 7

4 Data Structure **8**
4.1 Task 1 . . . 8

1 Objectives

1.1 Task 1

1. Take an input and verify that it is 8 or 7 numerical digits
2. Calculate the 8th check digit:

a Multiply the first 7 numbers alternately by 3,1

b Total these results

c Subtract this sum from its nearest highest multiple of 10

3. Compare this to the given 8th number, or complete the 7-digit number

1.2 Task 2

1. Take an input and validate that it is 8 numerical digits
2. Connect to a SQL database and run a query
3. Collect and display the results
4. Update the database with the customer?s order
5. Print a receipt
6. Cope with SQL errors

1.3 Task 3

1. Scan a database and find stock to order
2. Create a receipt of the order
3. Update the database with the updated stock level
4. Cope with any SQL errors

2 Test Plan

2.1 Task 1

1. Input strings of incorrect length. If rejected, it passes.

   Test data: `12345`

   Test data: `1234567890`

2. Input strings of letters. If rejected, it passes.

   Test data: `abcdefg`

3. Get program to run a valid input. Print out totals at each stage, and check them manually. If they are the same, it passes.

   Test data: `13245627`

i. Manually add the totals of a valid input. If they are the same, it passes.

ii. Manually round the total to the highest 10. If it is the same, it passes.

iii. Manually collect the distance rounded. If it is the same, it passes.

4. Run the program with a GTIN number taken from a product. If it correctly calculated and verified, it passes.

2.2 Task 2

1. Input strings of incorrect length. If rejected, it passes.

   Test data: `12345`.

   Test data: `1234567890`.

2. Input strings of letters. If rejected, it passes.

   Test data: `abcdefg`.

3. Input a valid string to search. If product found, it passes.

   Test data: `11440529`.

4. Manually check that the program has displayed the correct stock level and information. If it does, it passes.

   Stock info: 50 in stock for #`11440529` (red paint 100ml).

5. Order a quantity of the product. If the program updates the stock levels, it passes.

   Test data: order 5 x QTY of #`11440529` (red paint 100ml).

6. Complete a full order. If the program displays a receipt with the correct values, it passes.

   Test data: order 5 x QTY of #`11440529` (red paint 100ml) AND

   6 x QTY of #`11509493` (blue paint 100ml).

   Expected result: 5 x #`11509493` = £9.95 AND 6x #`11509493` = £11.94, total: £21.90 .

7. Provide the program with invalid values (such as ordering negative values). If the program rejects these, it passes.

   Test data: -3 x QTY #`11509493`.

   Expected result: error and re-input.

   Test data 0 x QTY #`11509493`.

Expected result: error and re-input.

Test data: 100 x QTY #`11509493`.

Expected result: not enough stock, re-input.

2.3 Task 3

1. Edit database for reduced stock. If reduced stock is identified, it passes.

   Test data: Edit a stock level to -10 of stock level.

   Print receipt.

2. Edit database for reduced stock. If a human-readable receipt is produced, it passes.

   Test data: Edit two stock levels to -10 of stock level.

   Print receipt.

3. Complete order for reduced stock. If the database updated, it passes.

   Continue from test area 2, and update database.

   Check database manually.

4. If the program handles SQL errors, it passes.

   Attempt to update from more than stock level.

3 Pseudocode

3.1 Task 1

```
START
User INPUT choice for calculate or verify
IF Calculate:
Number Length is 7
  INPUT GTIN
  CALL Verify function
ELSE IF Verify:
  Number Length is 8
  INPUT GTIN
  CALL verify Function
ENDIF
ENDIF
Verify Function:
  IF GTIN length = Length AND is all numeric:
    For a loop of 7 by step of 2:
      total = total+(GTIN [counter]*3)
      IF counter =6:
        Round total UP to nearest multiple of 10
        result = roundedNumber - total
          If Length = 7:
            Print result
          ELSE
            If GTIN at position Length = result:
              Print GTIN is a valid number
            ELSE:
              Print GTIN is an invalid number
            ENDIF
      ELSE
        Multiply GTIN at position of counter+1 by 1 and add to total
      ENDIF
  ELSE
    Print error and return to GTIN input
  ENDIF
END
```

3.2 Task 2

```
START
User INPUT GTIN number
IF GTIN is numerical AND GTIN = 8 charachters:
  Search Database for GTIN number
  IF result found:
    User INPUT quantity to order
    IF quantity > 0 AND quantity <= stock available
      PRINT receipt with total cost (cost per item * quantity ordered)
      Update Database with new stock (stock available a quantity ordered)
      User INPUT choice to order more items
      IF choice = yes:
        Add to order list and return to GTIN INPUT
      ELSE
        PRINT final receipt (order list) and end program
      ENDIF
    ELSE
      Return to Quantity INPUT
    ENDIF
  ELSE
    Return to GTIN INPUT
  ENDIF
ELSE
  Return to GTIN INPUT
ENDIF
END
```

3.3 Task 3

```
START
Connect to SQL Database
Search database:
  IF stock level < target stock level:
  RETURN results
PRINT results
APPEND results to order list
IF order complete:
  UPDATE database:
  stock level = target stock level
ELSE
  END
ENDIF
END
```

4 Data Structure

4.1 Task 1

Variable Name	Variable Description
`ask`	The choice wether the user wants to verify or calculate
`gtin`	The GTIN number used
`length`	The length the GTIN should be
`total`	The running total of all the multiplications
`checkdig`	The 8th digit in a verification
`rounded`	`total` Rounded up to the nearest 10
`result`	The 8th digit as the program calculates it
`again`	The choice of wether the user wants to run the program again

4.2 Task 2

Variable Name	Variable Description	Value
`con` and `cur`	Connections to SQL database	N/A
`var`	User Input GTIN number	User Defined
`results`	Fetchall results from SQL query	N/A (list)
`product`	Equal to `results`, reformatted	Equal to `results`
`sizeName` and `sizeNameRaw`	Variables used to format the name of the product	Name of product selected
`QtyToOrder`	User Input quantity ordered	User Defined
`NewStockAvab`	Variable used to update the SQL database with the new stock levels	Stock Available minus `QtyToOrder`
`costOfOrder`	Total cost of order	Price of product*`QtyToOrder`
`currentOrderAddRaw` and `currentOrderAdd` and `currentOrder`	Variables used to format and append the order to the entire list of order (to print receipt)	N/A