


Part 2

Pupils from the Drumrunie Academy Enterprise Club have been selling calendars. They distributed order forms numbered 1 to 90 to the three groups below who have been taking orders from customers.

Group Number	Order Form Numbers
1	1 – 30
2	31 – 60
3	61 – 90

An example of an order form is shown below.

ORDER FORM NUMBER ... 4		
Customer Name	Qty	Total
Mrs Smith	3	£13.50
Mr Khan	1	£4.50
Miss Green	2	£9.00
ORDER VALUE		£27.00



The Enterprise Club would like to calculate their total earnings from the fundraising event. Your task is to write a program that will take in the order form number and the order value using the following data:

Order Form Number	Order Value (£)
1	40.50
43	90.00
78	13.50
27	49.50
4	27.00
90	31.50

The program will then calculate the individual group totals and the overall total raised by the Enterprise Club.

The program requires the following inputs:

- How many order forms have been received and are ready to process
- A valid order form number
- The value of each order

The output from the program should display each group number and the total raised by each group. It should also display the overall total. All totals should be displayed to two decimal places. An example of the output is provided below:

Group Number	Order Value (£)
1	£117.00
2	£90.00
3	£45.00

The total raised for the project is £252.00

Your task is to create software for the project.

- The top level algorithm is shown below. Steps 5, 7 and 8 have been refined for you.

Pseudocode

MAIN STEPS

1. Initialise arrays to set 3 group totals
2. Get the quantity of orders to be processed
3. Loop for quantity of orders
4. Get a valid order form number
5. Process orders
6. End loop
7. Display group number and amount raised by group
8. Calculate and display overall total raised

REFINEMENTS

5. Process Orders
 - 5.1 Get order value
 - 5.2 Increment the total for the appropriate group

7. Display group number and amount raised by group
 - 7.1 Loop for number of groups
 - 7.2 Display group number and group total
 - 7.3 End loop

8. Calculate and display the overall total raised
 - 8.1 Overall total = group 1 total + group 2 total + group 3 total
 - 8.2 Display overall total

Tasks		Evidence required														
1	Refine the following parts of the algorithm: <ul style="list-style-type: none"> • Get a valid order form number (step 4) • Increment the total for the appropriate group (step 5.2) (NOTE: <i>all refinements must include an algorithm and not simply use a feature of an event-driven language.</i>)	Pseudocode for steps 4 and 5.2														
2	Create a program that matches the refined algorithm.	Listing of program														
3	Test your program using the data for the six orders detailed below. <table border="1" data-bbox="261 656 606 936"> <thead> <tr> <th>Order Number</th> <th>Order Value (£)</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>40.50</td> </tr> <tr> <td>43</td> <td>90.00</td> </tr> <tr> <td>78</td> <td>13.50</td> </tr> <tr> <td>27</td> <td>49.50</td> </tr> <tr> <td>4</td> <td>27.00</td> </tr> <tr> <td>90</td> <td>31.50</td> </tr> </tbody> </table>	Order Number	Order Value (£)	1	40.50	43	90.00	78	13.50	27	49.50	4	27.00	90	31.50	Printed output
Order Number	Order Value (£)															
1	40.50															
43	90.00															
78	13.50															
27	49.50															
4	27.00															
90	31.50															
4	Create another set of test data that will demonstrate how your program will respond to an invalid order number being entered.	Printed output Screenshots														
5	Evaluate the test data from Task 3 and Task 4 in terms of fitness for purpose.	Report														