Answers to Chapter 10 questions

Activity 10.1

```
Amount       100
TotalPrice   350
Discount     0.2
FinalPrice   280
Name         "Nikki"
Message      "Hello Nikki"
```

Activity 10.2

```
INPUT PercentageMark
      IF PercentageMark < 0 OR PercentageMark > 20
        THEN PRINT "Invalid Mark"
        ELSE
        IF PercentageMark > 9
          THEN PRINT "Pass"
          ELSE PRINT "Fail"
        ENDIF
      ENDIF
Normal data - 4
Boundary data 0, 20 accepted -1, 21 rejected
Erroneous data -11, a
```

Activity 10.3

```
CASE Day OF
  1 : PRINT "Sunday"
  2 : PRINT "Monday"
  3 : PRINT "Tuesday"
  4 : PRINT "Wednesday"
  5 : PRINT "Thursday"
  6 : PRINT "Friday"
  7 : PRINT "Saturday"
  OTHERWISE PRINT "Error"
ENDCASE
```

Activity 10.4

a

```
Total ← 0
FOR Counter ← 1 to 10
  INPUT Number
  Total ← Total + Number
NEXT
Average ← Total/10
```

b Using –1 to finish:

```
Total ← 0
Counter ← 1
INPUT Number
REPEAT
  Total ← Total + Number
  Counter ← Counter + 1
  INPUT Number
UNTIL Number = -1
Average ← Total/Counter
```

c Why the loop structures were chosen:
`FOR ... NEXT` loop as the number of repeats is known, 10.
`REPEAT ... UNTIL` loop as number of repeats is not known and an ending condition can be tested.

Activity 10.5

0, 26	Erroneous
1, 25	Boundary
9, 10	Boundary
19, 20	Boundary

More data: 2, 17, 22 Expected results 40, 306, 352

Activity 10.6

a Changes are shown in red.

```
OverallHighest ← 0
OverallLowest ← 100
OverallTotal ← 0
FOR Test ← 1 to 2
  SubjectHighest ← 0
  SubjectLowest ← 100
  SubjectTotal ← 0
  CASE Test OF
    1 : SubjectName ← "Maths"
    2 : SubjectName ← "Science"
    3 : SubjectName ← "English"
    4 : SubjectName ← "IT"
    OTHERWISE
  ENDCASE
  FOR StudentNumber ← 1 TO 5
    REPEAT
      PRINT "Enter Student", StudentNumber, "'s mark for ",
      SubjectName
     INPUT Mark
    UNTIL Mark < 101 AND Mark > -1
    IF Mark < OverallLowest THEN OverallLowest ← Mark
    IF Mark < SubjectLowest THEN SubjectLowest ← Mark
    IF Mark > OverallHighest THEN OverallHighest ← Mark
    IF Mark > SubjectHighest THEN SubjectHighest ← Mark
    OverallTotal ← OverallTotal + Mark
    SubjectTotal ← SubjectTotal + Mark
  NEXT
  SubjectAverage ← SubjectTotal/5
  PRINT SubjectName
  PRINT "Average is ", SubjectAverage
  PRINT "Highest Mark is ", SubjectHighest
  PRINT "Lowest Mark is ", SubjectLowest
NEXT
OverallAverage ← OverallTotal/10
PRINT "Overall Average is ", OverallAverage
PRINT "Overall Highest Mark is ", OverallHighest
PRINT "Overall Lowest Mark is ", OverallLowest
```

b Test data:

- Data set 1: 20, 30, 40, 50, 60, 10, 30, 40, 50, 70 — normal data
- Data set 2: 0, 0, 0, 0, 0, 100, 100, 100, 100, 100 — accepted boundary data
- Data set 3: –1, 101 — rejected boundary data
- Data set 4: A, –20, 190 — erroneous data

c Using data set 1:

Overall Highest	Overall Lowest	Overall Total	Test	Subject Highest	Subject Lowest	Subject Total	Student Number	Mark	Subject Average	Overall Average
0	100	0	1	0	100	0				
20	20	20		20	20	20	1	20		
30		50		30		50	2	30		
40		90		40		90	3	40		
50		140		50		140	4	50		
60		200		60		200	5	60	40	
			2	0	100	0				
	10	210		10	10	10	1	10		
		240		30		40	2	30		
		280		40		80	3	40		
		330		50		130	4	50		
70		400		70		200	5	70	40	40

Activity 10.7

Drawing a flowchart is going to be problematic for the following reasons:

- difficult to fit meaningful variable names in flowchart boxes
- difficult to show three nested loops in a clear way
- long and complex routines may not fit on a single page, and since flowcharts are a pictorial representation they do not work well over multiple pages.

End-of-chapter questions

1

```
IF … THEN … ELSE … ENDIF
CASE … OF … OTHERWISE … ENDCASE
```

2 **a** CASE statement

b A number of specific choices each with a different outcome and a need to reject values other than those choices.

c

```
Flag ← 0
REPEAT
  PRINT "                Menu"
  PRINT "        1 Set up new Account"
  PRINT "2 Make changes to an Existing Account"
  PRINT "         3 Close an Account"
  PRINT "         4 View my Orders"
  PRINT "        5 Place a New Order"
  PRINT "6 Alter an existing Order"
  PRINT "0 Exit                          H Help"
  INPUT Choice
  CASE Test OF
    1 : SetUpAccount
    2 : ChangeAccount
    3 : CloseAccount
    4 : ViewOrders
    5 : PlaceOrder
    6 : AlterOrder
    0 : Exit
    H : Help
    OTHERWISE : Flag ← 1
  ENDCASE
UNTIL Flag = 0
```

d Data set 1: 1, 5, H — normal data
Data set 2: 7, 11, a — rejected erroneous data

3 1st way:

```
Total ← 0
Counter ← 0
PRINT "Enter value"
Input Value
WHILE Counter < 5 DO
  Total ← Total + Value
  PRINT "Enter value"
  Input Value
  Counter ← Counter + 1
ENDWHILE
Print Total
```

2nd way:

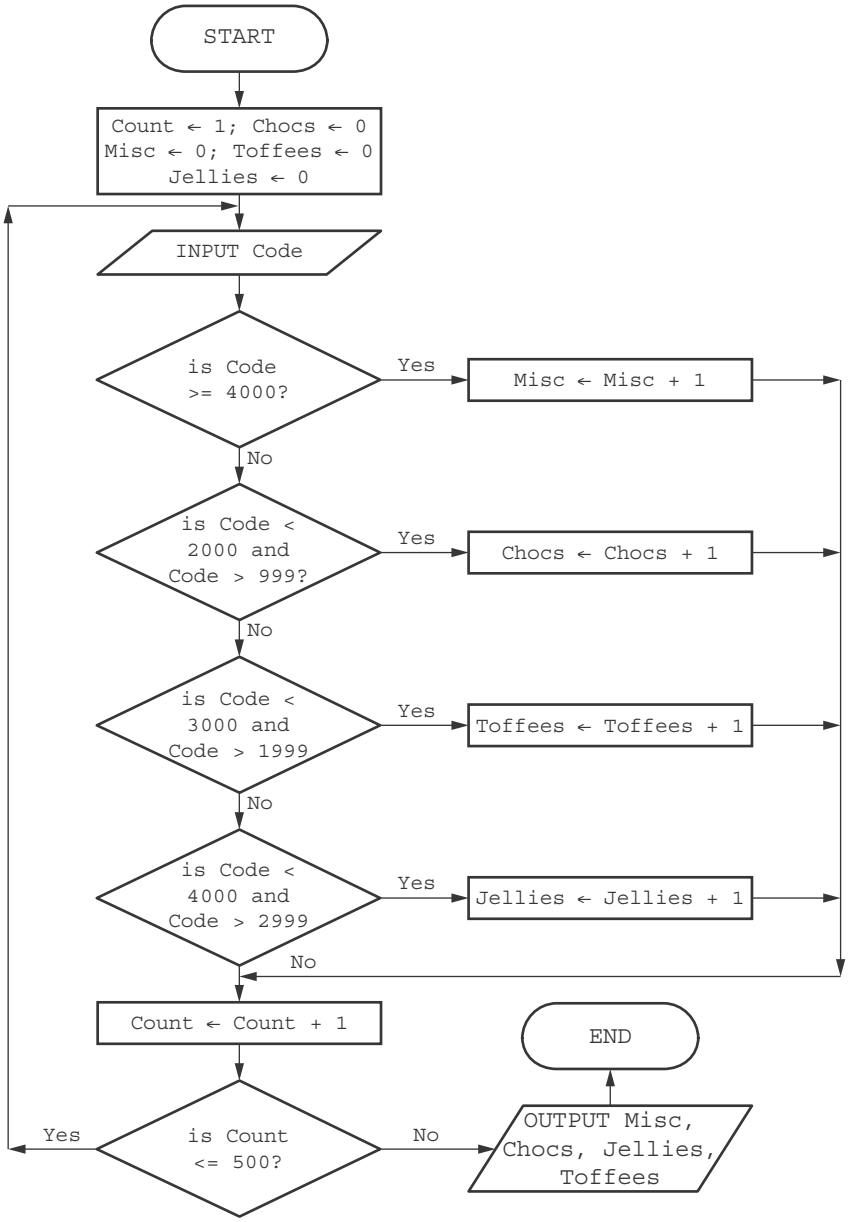
```
Total ← 0
Value ← 0
Counter ← 0
REPEAT
  Total ← Total + Value
  PRINT "Enter value"
  Input Value
  Counter ← Counter + 1
UNTIL Counter >= 5
Print Total
```

3rd way:

```
Total ← 0
FOR Counter ← 1 TO 5
  PRINT "Enter value"
  Input Value
  Total ← Total + Value
NEXT
Print Total
```

The most efficient is the `FOR ... NEXT` loop as the number of repeats is known.

4 a



b Use 10 rather than 500 different sorts of sweets.

c Test data: 4220, 5120, 1830, 2750, 3460, 7120, 3000, 2000, 1000, 8000

Code	Counter	Misc	Chocs	Jellies	Toffees	Output
	1	0	0	0	0	
4220	2	1				
5120	3	2				
1830	4		1			
2750	5				1	
3460	6			1		
7120	7	3				
3000	8			2		
2000	9				2	
1000	10		2			
8000	11	4				4, 2, 2, 2

5

```
GetTemperature
IF Temperature >= 22
  THEN FanOn
  ELSE
  IF Temperature <= 16
    THEN HeaterOn
    ELSE
      BEGIN
        FanOff
        HeaterOff
      END
  ENDIF
ENDIF
```

6 a

```
INPUT CountryName
INPUT H, M
IF CountryName = "Mexico"
  THEN H ← H - 7
  ELSE IF CountryName = "India"
    THEN
    H ← H + 4
    M ← M + 30
    ELSE IF CountryName= "New Zealand"
      THEN H ← H + 11
      ELSE PRINT "Error"
    ENDIF
  ENDIF
ENDIF
```

b Set 1: India, 3 hours 20 minutes; normal data that should work
Set 2: France, 418 hours 35 minutes; data that will cause erroneous output

7

```
TotalHeight ← 0
Total Weight ← 0
FOR Student ← 1 TO 1000
  INPUT Height
  WHILE Height > 2 OR Height < 0
    DO PRINT 'Error, please enter height again'
    INPUT Height
  ENDWHILE
  INPUT Weight
 WHILE Weight > 130 OR Weight < 0
    DO PRINT 'Error, please enter weight again'
    INPUT Weight
  ENDWHILE
  TotalHeight ← TotalHeight + Height
  Total Weight ← TotalWeight + Weight
NEXT Student
AverageHeight ← TotalHeight / 1000
AverageWeight ← TotalWeight / 1000
PRINT AverageHeight, AverageWeight
```

8

```
HighTake ← 0
TotBun ← 0
TotCoffee ← 0
TotCake ← 0
TotSand ← 0
TotDessert ← 0
REPEAT
 INPUT Item
 IF item = 'bun' THEN TotBun ← TotBun + 0.5
  ELSE IF Item = 'coffee' THEN TotCoffee ← TotCoffee + 1.20
   ELSE IF Item = 'cake' THEN TotCake ← TotCake + 1.50
    ELSE IF Item = 'sandwich' THEN TotSand ← TotSand + 2.10
     ELSE IF Item = 'dessert' THEN TotDessert ← TotDessert + 4.00
      ELSE PRINT 'error'
     ENDIF
    ENDIF
   ENDIF
  ENDIF
 ENDIF
UNTIL Item = 'end'
IF TotBun > HighTake THEN HighTake = TotBun
IF TotCoffee > HighTake THEN HighTake = TotCoffee
IF TotCake > HighTake THEN HighTake = TotCake
IF TotSand > HighTake THEN HighTake = TotSand
IF TotDessert > HighTake THEN HighTake = TotDessert
Total = TotBun + TotCoffee + TotCake + TotSand + TotDessert
PRINT Total, HighTake
```

9

```
Single ← 0
Two ← 0
Three ← 0
Four ← 0
Error ← 0
FOR Counter ← 1 to 5000
  INPUT Number
  IF Number > 999
    THEN Four ← Four + 1
    ELSE IF Number > 99
      THEN Three ← Three + 1
      ELSE IF Number > 9
       THEN Two ← Two + 1
       ELSE IF Number > 0
        THEN Single ← Single + 1
        ELSE Error ← Error + 1
       ENDIF
      ENFIF
    ENDIF
  ENDIF
NEXT Counter
Percent = Error / 50
PRINT Single, Two, Three, Four, Percent
```