

Pseudocode

There are 3 programming/pseudocode constructs:

- 1. Sequence:** It refers that instructions should be executed one after another.
- 2. Selection:** This construct is used to make a decision in choosing an option from many available options on the basis of a condition. So, if a condition is true then one option would be chosen while if a condition is false then another option will be chosen.
- 3. Repetition:** This construct is used to repeat a block of code as per the given condition.

Algorithm is step by step solution to a given problem. It is written in plain English statements. Algorithm is usually transformed into pseudocode or program flowchart.

Once the algorithm is tested in pseudocode or program flowchart then finally it is written in a specific programming language.

Pseudocode is a false code which consists of plain English statements, mathematical notations and keywords that are commonly found in high level languages.

It does not follow strict rules and style of any particular programming language. Pseudocode is used for learning programming concepts and to describe ideas before coding begins.

ARITHMETIC OPERATORS: In pseudocode arithmetic operators are used to perform arithmetic operations. These operators are listed below:

Arithmetic Operator	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division
^	Show power of a number

COMPARISION OPERATORS: These operators are used to compare different values.

Operator	Comparison
>	greater than
<	less than
=	equal
>=	greater than or equal
<=	less than or equal
<>	not equal
()	group
AND	both
OR	either
NOT	not

Assignment Operator:

Assignment operator is used to assign the value or expression to a variable. The value or expression is on the right side of assignment operator while the variable is on the left side of the assignment operator.

It is denoted by either of the following:

- ←
- =
- :=

For eg:

num1 ← 5

num2 ← 10

Sum ← num1 + num2

Sum = num1 + num2

Sum := num1 + num2

Note:

In CIE Exam, mostly ← sign is used for assignment operator.

Examples of pseudocode assignments:

Cost ← 10

Cost has the value 10

Price ← Cost * 2

Price has the value 20

Tax ← Price * 0.12

Tax has the value 2.4

SellingPrice ← Price + Tax

SellingPrice has the value 22.4

Gender ← "M"

Gender has the value M

Chosen ← False

Chosen has the value False

INPUT in Pseudocode:

In pseudocode we indicate the operation of taking input from users by either of the following keywords:

- **INPUT**
- **READ**
- **ENTER**

OUTPUT:

In pseudocode we indicate the operation of displaying a value or an output or any message by using either of the following keywords:

- **OUTPUT**
- **WRITE**
- **PRINT**

Note:

In CIE exam, mostly **INPUT, OUTPUT** and **READ, WRITE** keywords are used.

VARIABLE:

It is a named memory space which is used to store values.

Variable usually stores two kind of information:

1. All the input values from user must be stored in a variable
2. Result of some mathematical operation must be stored in a variable.

These are the rules while assigning name to a variable:

- There is no space in a variable name
- Variable name should not be a keyword of pseudocode
- Variable name should be relevant.

Example 1: Write Pseudocode that will take two numbers as input, calculates their sum and displays output.

Solution:

WRITE "Please enter two numbers to add"

READ num1

READ num2

Sum \leftarrow num1+num2

WRITE Sum

or

WRITE "The Answer is: ", Sum

In the above written pseudocode, num1 and num2 are the variable names where the two input number given by user will get store. The sum of num1 and num2 is a mathematical operation and the answer is saved in a variable named as sum. It is not necessary that name of variable should be same but it should be relevant. For example, the variable name Sum could be answer, ans, result etc.

Example 2: Write Pseudocode that will take two numbers as input, calculates their product and displays output.

Solution: (Solve at your own)

Example 3: Write down Pseudocode that will take marks of physics, chemistry and math as input, calculates the average and displays output.

Solution:

```
WRITE "please enter marks of physics"  
READ Phy_marks  
WRITE "please enter marks of chemistry"  
READ Chem_marks  
WRITE "please enter marks of maths"  
READ math_marks  
Avg ← (Phy_marks + Chem_marks + math_marks)/3  
WRITE Avg
```

TOTALLING & COUNTING:

Totaling is a process to add up the series of number. A variable named as total or sum is used to hold on to the running total. So

$Total \leftarrow Total + Number$

It actually means:

$Total(New\ Value) \leftarrow Total(Old\ Value) + Value\ of\ Number$

Note: To perform the totaling, total variable must be $total \leftarrow 0$ at the start of pseudocode.

Counting is a process to count how many times something happens.

Count \leftarrow Count + 1

Count \leftarrow Count - 1

Count \leftarrow Count + 5

A variable named as count is used for counting purposes.

Sometimes, it is also represented as

Count Incremented By 1

or

Count Decremented By 1

REPETITION:

The process of repeating a block of pseudocode is called as repetition. It is also known as **looping** or **iteration**.

There are three types of repetition statements

- 1. FOR ... TO NEXT**
- 2. WHILE... DO ... ENDWHILE**
- 3. REPEAT ... UNTIL**

FOR... TO ... NEXT

This repetition statement is used when we know how many times an instruction or set of instructions is to be repeated.

Few things to remember about FOR...TO...NEXT loop are:

- There is a variable in FOR ... TO ... NEXT loop for controlling the number of iterations and is known as a control variable. The name of the control variable is usually 'Count'.
- We specify the initial (lower) and final (higher) values of the control variable in the opening statement of the loop. These initial and final values are not restricted to the numerical values only, they can be variables as well.
- The Control Variable is automatically incremented by '1' each time the loop ends.
- The value of the control variable is tested at the beginning of the loop & loop is repeated until the value of control variable is less than or equal to the specified final value.

Example 4: Write pseudocode that will take 10 numbers as input and print their average by using FOR...TO...NEXT loop.

Solution:

```
Total ← 0
FOR Count ← 1 TO 10
WRITE "Enter number"
READ num
Total ← Total+num
NEXT
avg ← Total/10
WRITE "Average of 10 Numbers is:", Total
```

Few things to Remember:

- Whenever **average** is to be calculated then **totaling** must be performed.
- Assign **total ← 0** always. As in the above example it is assigned 0.
- Take care of the initial and final limits of count in the condition of a loop. For example, **FOR Count ← 1 TO 10** will repeat the code for 10 times but **FOR Count ← 0 TO 10** will repeat the code for 11 times.
- FOR Loop must be terminated by the **NEXT** keyword.

Example 5: A geography class decided to measure daily temperature and hours of sunshine per day over a 12 months period (365 days). Write pseudocode that inputs the temperature and hours of sunshine for all 365 days and give output as average temperature for the year and avg. number of hours per day over the year.

Solution:

```
Total_temp ← 0
Total_Shrs ← 0

FOR Count ← 1 to 365
WRITE "Please enter temperature"
READ temp
Total_temp ← Total_temp + temp
WRITE "Please enter sunshine hours"
READ Shours
Total_Shrs = Total_Shrs + Shours
NEXT

Avgtemp ← Total_temp/365
AvgShrs ← Total_Shrs/365

WRITE "Average temperature for whole year is:", Avgtemp
```

WRITE "Average Sunshine hours for whole year is:", AvgShrs

Example 6: Mr. John wants to calculate the average marks for his class of 28 students for Urdu, Islamiat and Pak.studies. Write down pseudocode that will help him to accomplish this task.

Solution: (Solve at your own)

WHILE ... DO ... ENDWHILE

This repetition statement is used when we don't know how many times an instruction or set of instructions is to be repeated.

Few things to remember about WHILE...DO...ENDWHILE loop are:

- The loop is repeated until a condition is true and halted when the condition is false.

- Condition is tested at the beginning of the loop.
- The statements/instruction between DO and ENDWHILE keywords are repeated.

Example 7: Write pseudocode that will take numbers input and add them while the input number is greater than or equal to 0. Print the final result.

Solution

Total \leftarrow 0

WRITE "Input Number"

READ num

WHILE num \geq 0

DO

Total \leftarrow Total + num

READ num

ENDWHILE

WRITE "Total sum is:", Total

Example 8: Write pseudocode that will take numbers input, add them and calculates the average while the input number is greater than or equal to 0. Print the average of all input numbers.

Solution:

Total \leftarrow 0

Count \leftarrow 1

WRITE "Input Number"

READ num

WHILE num \geq 0

DO

Total \leftarrow Total + num

READ num

Count \leftarrow Count + 1

ENDWHILE

avg \leftarrow Total/Count

WRITE "The Average is:", avg

Note: Although there is a difference of operation between FOR...TO...NEXT Loop and WHILE...DO...ENDWHILE Loop but still there is a possible way to rewrite a code written in FOR...TO...NEXT loop into WHILE...DO...ENDWHILE Loop.

Example 9: Rewrite the pseudocode given below by using WHILE...DO...ENDWHILE loop.

```
Total ← 0
FOR Count ← 1 TO 10
WRITE "Enter number"
READ num
Total ← Total+num
NEXT
avg ← Total/10
WRITE "Average of 10 Numbers is:", Total
```

Solution:

```
Total ← 0
Count ← 1

WHILE Count <= 10
DO
WRITE "Enter number"
READ num
Total ← Total+num
Count ← Count + 1
ENDWHILE
avg ← Total/10
WRITE "Average of 10 Numbers is:", Total
```

Example 10: Rewrite the pseudocode given below by using FOR...TO...NEXT loop.

```
Total ← 0
Count ← 1

WHILE Count <= 15
DO
WRITE "Enter a Number"
READ num
Total ← Total + num
Count ← Count + 1
ENDWHILE

avg ← Total/Count
WRITE "The Average is:", avg
```

Solution:

```
Total ← 0

FOR Count ← 1 TO 15
WRITE "Enter a Number"
```

```
READ num
Total ← Total + num
NEXT
```

```
avg ← Total/Count
WRITE "The Average is:", avg
```

Example 11: A pseudocode is given below which will take age of 18 students and print out the min age of students. Rewrite all the pseudocode using WHILE...DO...ENDWHILE loop.

```
WRITE "Enter age of student"
min_age ← 100
FOR COUNT ← 1 TO 18
READ age
IF age < min_age
THEN
min_age ← AGE
ENDIF
NEXT
WRITE "Minimum AGE:" min_age
```

Solution: (Solve it at Your Own)

Example 12: Rewrite the code written below using WHILE...DO...ENDWHILE Loop.

```
Total_temp ← 0
Total_Shhs ← 0
```

```
FOR Count ← 1 to 365
WRITE "Please enter temperature"
READ temp
Total_temp ← Total_temp + temp
WRITE "Please enter sunshine hours"
READ Shours
Total_Shrs = Total_Shrs + Shours
NEXT
```

```
Avgtemp ← Total_temp/365
AvgShrs ← Total_Shrs/365
```

```
WRITE "Average temperature for whole year is:", Avgtemp
WRITE "Average Sunshine hours for whole year is:", AvgShrs
```

Solution:

REPEAT ... UNTIL

It is a repetition statement that is used when we don't know how many times an instruction or set of instructions is to be repeated.

- It is different from WHILE... DO ... ENDWHILE because this loop will be repeated until a condition is false and it will stop executing once the condition is true.

- The condition is tested at the end of the loop and even if a condition is true the loop will execute at least once.

Example 13: Write the pseudocode that will take numbers as input, add and gives the total as output. The loop will continue until "0" is given as input.

Solution

```
Total ← 0
REPEAT
WRITE "Enter a number to add"
READ num
Total = Total + num
UNTIL num is "0"
Write Total
```

Example 14: write a pseudocode that will take anything as input and will display it as output until A or B is pressed

Solution

```
REPEAT
WRITE "enter a number, value, character or symbol to display"
READ anyvalue
UNTIL anyvalue is "A" or "B"
```

SELECTION STATEMENTS:

These are also known as conditional statements.

There are two types of selection statements:

1. **IF ... THEN... ELSE ... ENDIF Statements:**
2. **CASE ... OF ... OTHERWISE ... ENDCASE Statements**

IF ... THEN... ELSE... ENDIF:

The selection statement is used when we want to perform one operation when a condition is true and another operation when a condition is false, or another condition is true.

It is used to choose a route from all available routes in an algorithm/pseudocode.

Example 15: Write pseudocode that will take a number as input and tells whether a number is positive, negative or zero.

Solution:

```
WRITE "Enter a number"
READ num
IF num > 0
  THEN
    WRITE "The number is positive"
```

```
ELSE IF num = 0
  THEN
    WRITE "The number is zero"
ELSE
  WRITE "The number is negative"
ENDIF
ENDIF
```

Example 16: Write pseudocode that will take marks in an exam as input and will tell the grade A* for 90 to 100, A for 80 to 89, B for 70 to 79, C for 60 to 69 and F grade for 0 to 59.

```
WRITE "Enter Marks"
READ marks
  IF marks >= 90 AND marks <= 100
    THEN
      WRITE "The Grade is A*"
    ELSE IF marks >= 80 AND marks < 90
      THEN
        WRITE "The Grade is A"
  ELSE IF marks >= 70 AND marks < 80
    THEN
      WRITE "The Grade is B"
  ELSE IF marks >= 60 AND marks < 70
    THEN
      WRITE "The Grade is C"
  ELSE IF marks >= 0 AND marks < 60
    THEN
      WRITE "The Grade is F"
  ELSE
    WRITE "Invalid Marks"
  ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
```

Note: Each IF statement and ELSE IF statement must have an ENDIF statement to terminate.

Example 17: Write pseudocode that performs the following: Ask a user to enter a number. If the number is between 0 and 10, write the word blue. If the number is between 10 and 20, write the word red. if the number is between 20 and 30, write the word green. If it is any other number, write that it is not a correct color option.

Solution: (Solve at Your Own)

Example 18: Write down pseudocode that will take three numbers as input and tells

(a) The Largest Number

(b) The Smallest Number

Solution: (a)

WRITE "Enter three numbers"

READ num1

READ num2

READ num3

IF num1 > num2 AND num1 > num3

THEN

WRITE "The largest Number is", num1

ELSE IF num2 > num1 AND num2 > num3

THEN

WRITE "The largest Number is", num2

ELSE

WRITE "The largest Number is", num3

ENDIF

ENDIF

Solution: (b) Solve at Your Own

CASE ... OF ... OTHERWISE ... ENDCASE STATEMENT:

When there are too many available routes in an algorithm/pseudocode then it requires too many IF ... THEN ... ELSE ... ENDIF statements to make a selection among these routes which is not an easy task and it makes pseudocode difficult to manage. To overcome this issue CASE ... OF ... OTHERWISE Statement is used.

In short, CASE ... OF ... OTHERWISE ... ENDCASE is used in place of IF ... THEN ... ELSE... ENDIF statement when we have to make a selection of a route from too many available routes.

Example 19: Write down pseudocode that will take a number as input (from 1 to 7) and print the day name for corresponding number e.g 1 for Monday 2 for Tuesday and so on

Solution:

```
WRITE "ENTER A NUMBER TO DISPLY THE DAY NAME FOR IT"  
READ DAYNUM
```

```
CASE DAYNUM OF  
1: DAYNAME ← "MONDAY"  
2: DAYNAME ← "TUESDAY"  
3: DAYNAME ← "WEDNESDAY"  
4: DAYNAME ← "THURSDAY"  
5: DAYNAME ← "FRIDAY"  
6: DAYNAME ← "SATURDAY"
```

7: DAYNAME \leftarrow "SUNDAY"

OTHERWISE

WRITE "ERROR"

DAYNUM \leftarrow "UNKNOWN"

ENDCASE

WRITE "THE DAY IS:", DAYNAME

Example 20: write a pseudo code that will take two numbers as input perform any of the basic operation (+, -, *, /) as per user requires

Solution:

WRITE "ENTER TWO NUMBER"

READ NUM1

READ NUM2

WRITE "ENTER THE OPERATION"

READ OPRT

CASE OPRT OF

+ : ANS \leftarrow NUM1 + NUM2

- : ANS \leftarrow NUM1 - NUM2

/ : ANS \leftarrow NUM1 / NUM 2

* : ANS \leftarrow NUM1 * NUM2

OTHERWISE

WRITE : "INVALID OPERATION "

Ans \leftarrow "Invalide"

ENDCASE

WRITE: "Answer is: " , Ans

Practice Questions (Past Paper)

In CIE exam, mostly the pseudocodes are asked that can select maximum from many values, minimum from many values, can calculate average of given values, can count the number of digits in a given input or number and then performing various operations on its basis.

Question1:

(a) Write an algorithm, using pseudocode or a flowchart, which

- inputs a set of positive numbers (which end with -1)
- outputs the average (mean) value of the input numbers
- outputs the value of the largest (highest) number input

(b) Write an algorithm, using pseudocode or a flowchart, which

- inputs a whole number (which is > 0)
- calculates the number of digits in the number
- outputs the number of digits and the original number

(E.g. 147 would give an output of 3, 147)

(a) highest = -100; total = 0; count = 0 (1 mark) *initialise values NB highest cannot be 0*
input number (1 mark) *inputs in the correct place*
while number < > -1 **do** (1 mark) *loop until -1 is input*
 total = total + number (1 mark) *calculate number total*
 count = count + 1 *and count numbers input*
 if number > highest **then** highest = number (1 mark) *highest*
 input number
endwhile
average = total/count (1 mark) *calculate average value*
print average, highest *and output average and highest value*

[4]

(b) d = 0 (1 mark) *initialise value*
input number (1 mark) *input number and set variable to this number*
t = number
repeat (1 mark) *correct loop*
 t = t / 10 (1 mark) ***method to find number of digits*
 d = d + 1 (1 mark) ***counting number of digits*
until t < 1
print number, d (1 mark) *correct output outside the loop*
(** NOTE: there are other ways of finding number of digits e.g.
 if number > 0 **then** d = 1
 else if number > 9 **then** d = 2

 else if number > 999999 **then** d = 7 etc.)

If no loop then 0 for loop and 0 for output

[4]

Question 2:

17 A school is doing a check on the heights and weights of all its students. The school has 1000 students.

Write an algorithm, using pseudocode or a flowchart, which

- inputs the height and weight of all 1000 students
- outputs the average (mean) height and weight
- includes any necessary error traps for the input of height and weight

17 Marking Points:

- initialisation of running totals (1 mark)
 - correct loop control (1 mark)
 - error trap for height input (1 mark)
 - error trap for weight input (1 mark)
 - sum total1 and average1 (i.e. height) calculation (1 mark)
 - sum total2 and average2 (i.e. weight) calculation (1 mark)
 - correct output (only if some processing attempted, must be outside loop) (1 mark)
- [max: 5]

Sample pseudocode

total1 = 0: total2 = 0 (1 mark)

for x = 1 to 1000 (1 mark)

 input height, weight

 if height > 2 or height < 0 then print "error": input height (1 mark)

 if weight > 130 or weight < 0 then print "error": input weight (1 mark)

 else total1 = total1 + height: total2 = total2 + weight

next x

average1 = total1/1000 (1 mark)

average2 = total2/1000 (1 mark)

print average1, average2 (1 mark) [5]

Question 3:

16 (a) Write an algorithm, using pseudocode or a flowchart, which:

- inputs 50 numbers
- outputs how many of the numbers were > 100

16 (a) total = 0 (1 mark) *initialisation*
for x = 1 to 50 (1 mark) *correct loop*
input number (1 mark) *correct input and output*
if number > 100 then total = total + 1 (1 mark) *count numbers >100*

next x

output total

(1 mark for initialising total)

(1 mark for correct loop – accept **repeat** loop or a **while** loop)

(1 mark for correct input (within loop) and output (after the loop))

(1 mark for counting how many input numbers were > 100)

(b) Write an algorithm, using pseudocode or a flowchart, which:

- inputs 100 numbers
- finds the average of the input numbers
- outputs the average

(b) total = 0 (1 mark) *initialise total*
for x = 1 to 100 (1 mark) *correct loop*
input number (1 mark) *correct input and output*
total = total + number (1 mark) *finding sum of numbers*

next x

average = total/100 (1 mark) *calculate average*

output average

(1 mark for initialising total)

(1 mark for correct loop – accept **repeat** loop or a **while** loop)

(1 mark for correct input (inside the loop) and output (after the loop))

(1 mark for calculating total)

(1 mark for calculating the average outside the loop)

Question 4:

- 18 A group of students were monitoring the temperature every day over a one-year period. Readings were taken ten times **every** day (you may assume a year contains 365 days).

Write an algorithm, using pseudocode or flowchart, which

- inputs all the temperatures (ten per day)
- outputs the **highest** temperature taken over the year
- outputs the **lowest** temperature taken over the year
- outputs the average temperature **per day**
- outputs the average temperature **for the whole year**

18 Marking points (maximum of 7 marks)

- initialising highest and lowest to reasonable values (must **not** be zero)
- first loop controlling one year (365 days)
- re-setting total for **each** day
- second loop controlling readings taken **per day**
- read temperature
- calculate total day temperature
- calculate total year temperature
- identifying highest temperature
- identifying lowest temperature
- finding average temperature for day
- finding average temperature for year
- output average day temperature inside loop
- output highest, lowest, average outside the loop

Sample algorithm in pseudocode

```
highest = -100: lowest = 100: total year = 0           } 1 mark
for c = 1 to 365                                     } 1 mark
    total day = 0                                       } 1 mark
    for d = 1 to 10                                     } 1 mark
        read temp                                       } 1 mark
        total day = total day + temp                   } mark
        total year = total year + temp                 } 1 mark
        if temp > highest then highest = temp          } 1 mark
        if temp < lowest then lowest = temp            } 1 mark
    next d
    average day = total day/10                          } 1 mark
    print average day                                   } 1 mark
next c
average year = total year/3650                         } 1 mark
print highest, lowest, average year                   } 1 mark
```

Question 5:

17 (a) A car's speed is measured between points A and B, which are 200 km apart.



The final speed of the car is calculated using the formula:

$$\text{Final Speed} = \frac{200}{\text{Time (hours)}}$$

What is the final speed of a car if it takes 2 hours to get from A to B?

(b) Write an algorithm, using pseudocode or otherwise, which inputs the times for 500 cars, calculates the final speed of each car using the formula in part (a), and then outputs:

- the final speed for ALL 500 cars
- the slowest (lowest) final speed
- the fastest (highest) final speed
- the average final speed for all the cars.

17 (a) 100 (km/hr)

(b) Marking points

Initialisation (slowest = 1000 or an equivalent high value)

Correct loops structure and control

Input (in correct place)

Calculation of final speed using given formula in part (a) inside the loop

Output the final speed for ALL cars inside the loop

Calculation highest speed input

Calculation slowest speed input

Calculate the average (two parts to this calculation)

Final outputs (correct place + some form of processing done)

Sample program:

```
total = 0
highest = 0
slowest = 1000
for n = 1 to 500
    input time
    finalspped = 200/time
    print finalspped
    total = total + finalspped
    if finalspped > highest
        then highest = finalspped
    if finalspped < slowest
        then slowest = finalspped
next n
average = total/500
print average, highest, slowest
```

} 1 mark
}
} 1 mark
} 1 mark
} 1 mark
}
} 1 mark
}
} 1 mark
} 1 mark
} 1 mark

Question 6:

18 A small airport handles 400 flights per day from three airlines:

FASTAIR	(code FA)
SWIFTJET	(code SJ)
KNIGHTAIR	(code KA)

Each flight is identified by the airline code and 3 digits. For example FA 156.

Write an algorithm, using pseudocode or otherwise, which monitors the 400 flights into and out of the airport each day. The following inputs, processing and outputs are all part of the monitoring process:

- input flight identification
- calculate number of flights per day for **each** of the three airlines
- output the percentage of the total flights per day by **each** airline
- any validation checks must be included

18 marking points (1 mark per item up to the maximum of 5):

initialise fa, sj and ka to zero

correct loop

inputs (in correct place)

addition of number of flights per airline

any validation checks carried out

calculate percentages

outputs (in correct place and ONLY if some evidence of any attempt at processing)

sample program/algorithm

```
fa = 0; sj = 0; ka = 0; } 1 mark
for x = 1 to 400 } 1 mark
    input lettercode }
    input numbercode } 1 mark
    if lettercode = "FA" then fa = fa + 1 }
    if lettercode = "SJ" then sj = sj + 1 } 1 mark
    if lettercode = "KA" then ka = ka + 1 }
    else print "error" } 1 mark
next x
fapercent = fa/4 }
sjpercent = sj/4 } 1 mark
kapercent = ka/4 }
print fapercent, sjpercent, kapercent } 1 mark
```


Question 7:

19 The manufacturing cost of producing an item depends on its complexity. A company manufactures three different types of item, with costs based on the following calculations:

Item type 1: item cost = parts cost * 1.5

Item type 2: item cost = parts cost * 2.5

Item type 3: item cost = parts cost * 5.0

The company makes 1000 items per day.

Write an algorithm, using pseudocode, flowchart or otherwise, which

- inputs the item type and parts cost of each item
- outputs the item cost for each item
- calculates and outputs the average (mean) item cost per day (based on 1000 items being made).

19 Marking points

correct loop
correct inputs
check for type and calculate itemcost
action taken if type NOT 1, 2 or 3
calculate totalcost
calculate the average totalcost
both outputs in the correct place

Sample algorithm:

total cost = 0

for x = 1 **to** 1000 (1 mark)

input type, partcost (1 mark)

if type = 1 **then** itemcost = partcost * 1.5}

if type = 2 **then** itemcost = partcost * 2.5} (1 mark)

if type = 3 **then** itemcost = partcost * 5.0}

else print error (1 mark)

 totalcost = totalcost + itemcost (1 mark)

print itemcost

next x

average = totalcost/1000 (1 mark)

print average (1 mark)

[5]

Question 8:

19 Customers can withdraw cash from an Automatic Teller Machine (ATM).

- withdrawal is refused if amount entered > current balance
- withdrawal is refused if amount entered > daily limit
- if current balance < \$100, then a charge of 2% is made
- if current balance \geq \$100, no charge is made

Write an algorithm which inputs a request for a sum of money, decides if a withdrawal can be made and calculates any charges. Appropriate output messages should be included.

19 Sample algorithm:

input amount

if amount > balance **then** x = 1 (2 marks)

else if amount > daily limit **then** x = 1 (1 mark)

else x = 0

while x = 0

if balance < 100 **then** charge = 0.02 * amount (1 mark)

else charge = 0 (1 mark)

endwhile

if x = 1 **then print** "Sorry, withdrawal refused"

print charge (1 mark)

Marking points

1 mark for checking if amount > balance

1 mark for checking if amount > daily limit

1 mark for some way of testing if withdrawal will be refused (value of x in above)

1 mark for checking if balance < \$100...

1 mark ...for calculating 2% charge

1 mark for no charge if balance \geq \$100

2 marks for giving correct outputs

Question 9:

16 (a) Fuel economy for a car is found using the formula:

$$\text{Fuel Economy} = \frac{\text{Distance Travelled (km)}}{\text{Fuel Used (litres)}}$$

What would be the Fuel Economy of a car travelling 40 km on 10 litres of fuel?

(b) The Fuel Economy for 1000 cars is to be calculated using the formula in Question **16(a)**.

Write an algorithm, using pseudocode or otherwise, which inputs the Distance Travelled (km) and the Fuel Used (litres) for 1000 cars. The Fuel Economy for each car is then calculated and the following outputs produced:

- Fuel Economy for each car
- average (mean) Fuel Economy for all of the cars input
- the best Fuel Economy (i.e. highest value)
- the worst Fuel Economy (i.e. lowest value)

16 (a) $40/10 = 4$

(b) general marking points

initialising **best** and **worst** to sensible values
correct loop for 1000 cars
correct use of calculation given in part **(a)**
output economy for each car inside loop
determining best economy
determining worst economy
calculating mean economy for all cars
input data **and** output all three results (only award mark if some form of processing done)

sample program

total = 0, count = 0, best = 0, worst = 1000	1 mark
repeat	1 mark
input litres, distance	
economy = distance/litres	1 mark
print economy	1 mark
if economy > best then best = economy	1 mark
if economy < worst then worst = economy	1 mark
total = total + economy	
count = count + 1	
until count = 1000	
average = total/1000	1 mark
print average, best, worst	1 mark

Question 10:

- 19 A company has 5000 CDs, DVDs, videos and books in stock. Each item has a unique 5-digit code with the first digit identifying the type of item, i.e.

1 = CD
2 = DVD
3 = video
4 = book

For example, for the code 15642 the 1 identifies that it is a CD, and for the code 30055 the 3 identifies that it is a video.

Write an algorithm, using pseudocode or otherwise, that

- Inputs the codes for all 5000 items
- Validates the input code
- Calculates how many CDs, DVDs, videos and books are in stock
- Outputs the **four** totals.

19 General marking points:

loop – 1 mark
input in correct place – 1 mark
checks on code – 1 mark
correct use of **if/then/else** or **case** statements – 1 mark
increment all totals – 1 mark
error recognition/validation – 1 mark
correct output in correct place – 1 mark

Sample program 1:

```
set c, d, v, b = 0: set count = 0
repeat
    input code
    x = code/10000
    y = INT(x)
    if y = 1 then c = c + 1
        else if y = 2 then d = d + 1
        else if y = 3 then v = v + 1
        else if y = 4 then b = b + 1
        else print "error"
    count = count + 1
until count = 5000
print c, d, v, b
```

1 mark
1 mark
} 1 mark
}
} 2 marks
}
1 mark
1 mark

Sample program 2:

```
set c, d, v, b = 0: set count = 0
repeat
    input code
    if code >= 1000 and code < 2000 then c = c + 1
    else if code >= 2000 and code < 3000 then d = d + 1
    else if code >= 3000 and code < 4000 then v = v + 1
    else if code >= 4000 and code < 5000 then b = b + 1
    else print "error"
    count = count + 1
until count = 5000
print c, d, v, b
```

1 mark
1 mark
}
} 3 marks
}
1 mark
1 mark

(NOTE – OK to use statements such as *if code begins with a 1* as code checks)

