



2.3.3 Selection

May/June 2005

5 (c) (ii) Describe how procedures and the programming construct “selection” can be used to code a simple menu system for a user. [3]

Oct/NOV 2006

3. (a) INPUT A, B

IF B = 0 THEN C = A

ELSE C = A ÷ B

ENDIF

PRINT A, B

PRINT C

END

Write down the outputs produced by the algorithm if

(i) A = 8, B = 2

[2]

(ii) A = 6, B = 0

[2]

(b) An automatic fan is designed so that it turns on only when a person is in the room AND the temperature is above a set value (D).

The fan receives information from two sensors.

1. A motion sensor which returns a value (M) dependent upon a person being sensed in the room.
2. A thermistor (electronic thermometer) which returns the temperature in the room (T).

Produce an algorithm to control the fan. Your algorithm may be expressed in any form. [6]





2.3.3 Selection

Oct/NOV 2007

6. A washing machine is computer controlled.

The full cycle consists of:

- Check the door is shut
- Heat water with heater (H) to 80 degrees if hot wash has been selected, 40 degrees if a cool wash has been selected
- Run wash motor (M) for 20 minutes, checking water every 5 minutes to see if heater (H) should be turned on again
- Sound buzzer to signify that wash cycle is complete.

Note: M can be set to “on” or “off”.

Write an algorithm, in whatever form, to show the working of the washing machine.

[7]

May/June 2008

4. (a) The water level in a reservoir is controlled by a computer system.

During normal operation the water level (W) is between the high water (H) and low water (L) marks. At these times the input valve (I) and output valve (O) are both open.

If the level reaches H then the input valve is shut off until the level falls below H again.

If the level falls below L then the output valve is shut off until the level rises above L again.

If the level falls below L for more than 1 hour, the system sends an alarm signal to the operator.

Using the variables W,H,L,I,O produce an algorithm to control the water level in the reservoir.

[8]

Oct/NOV 2009. P11

The offices of a government department deal with local taxes in a city.

10. Part of the processing of the data is to calculate the amount of tax which needs to be paid.

- The person’s total income for the year is input to the system
- The first \$500 is not taxed
- The remainder is taxed at 10%

Either the tax to be paid is output from the system OR a message is output to say there is no tax to pay

(a) Using I to stand for the total income and T to stand for the tax to be paid, produce an algorithm which will take I as its input and then calculate the tax.

[5]





2.3.3 Selection

(c) Describe how the algorithm can be used to calculate the tax for each person in the city without it having to be rerun. [3]

May/June 2011. P23

Jodelle first produces her solution using pseudocode. She wants the password to be 'poppy', the name of her cat.

```
Attempt ← 1
```

```
REPEAT
```

```
INPUT Password
```

```
Attempt ← Attempt +1
```

```
UNTIL Password = "poppy" OR Attempt = 3
```

```
IF Password = "poppy"
```

```
THEN
```

```
OUTPUT "password correct"
```

```
ELSE
```

```
OUTPUT "no valid password entered"
```

```
ENDIF
```

Jodelle first produces her solution using pseudocode. She wants the password to be 'poppy', the name of her cat.

```
Attempt ← 1
```

```
REPEAT
```

```
INPUT Password
```

```
Attempt ← Attempt +1
```

```
UNTIL Password = "poppy" OR Attempt = 3
```

```
IF Password = "poppy"
```

```
THEN
```

```
OUTPUT "password correct"
```

```
ELSE
```

```
OUTPUT "no valid password entered"
```





2.3.3 Selection

ENDIF

1. (f) Jodelle wants to output a comment after each failed password entry.

If Attempt = 1 she wants to output "First try is wrong. Please try again the second try she wants to output "Password still wrong. One more chance"

On the third try she wants to output "No valid password entered"

In a high-level language write a program for this idea using a CASE or SELECT statement.

[4]

May/June 2012. P21/22

1 Anna wants to find out about her fellow students' reading habits. It will be part of her Literature coursework.

She will ask questions online, so starts by designing a screen layout. The first four questions will ask for:

- student's first name
- date of birth
- type of book they prefer (printed, audio-book or e-book)
- whether student reads novels (yes/no)

(d) Anna is to write a program to analyse the responses.

Using nested IF...THEN statements, complete the pseudocode to calculate the totals for each BookType (printed, audio-book or e-book).

REPEAT

 READ next record

UNTIL no more student records

[5]

Oct/NOV 2012 P22

Soni works for a software house which has been asked to design software for a car hire company, Super Cars.

1 Soni decides on the main tasks:

- enter car details
- enter car hire details for
 - hirer
 - car
 - payment





2.3.3 Selection

(d) The colour of each car is also recorded. The following pseudocode should validate the car colour.

```
INPUT Colour
IF (Colour = "blue") OR (Colour = "black") OR (Colour = "grey")
THEN
OUTPUT Colour" is valid"
ELSE
OUTPUT "Invalid"
ENDIF
```

State the output when the colour input is:

- (i) green
- (ii) Black
- (iii) grey

[3]

Oct/NOV 2012 P23

Soni works for a software house which has been asked to design software for Super Bikes, a company that specialises in hiring out motorbikes.

1 Soni decides on the main tasks:

- enter bike details
- enter bike hire details
 - bike specification
 - service history
 - hire rates

(c) The type of each bike is also recorded. The intention of this pseudocode is to allow only trial, scooter and road as valid bike types.

```
INPUT BikeType
IF ((BikeType = "trial") AND (BikeType = "scooter"))
OR (BikeType = "road")
THEN
OUTPUT BikeType" is valid"
ELSE
OUTPUT "Invalid"
ENDIF
```





2.3.3 Selection

State the output when the `BikeType` input is:

- (i) scrambler
- (ii) trial
- (iii) road

[3]

(d) Rewrite the line in the pseudocode in part (c) which needs changing in order to correctly validate bike types. [1]

(e) (i) All bike registrations are five characters long, and have the format `BKXXX`, where `X` is a digit.

For example, `BK572` is a valid registration.

In a high-level programming language write code that will validate the format of a registration input into the variable `BikeReg`. (Add extra line numbers if required.)

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

[10]

(ii) Explain why `487BK` is an invalid bike registration.

State the line number of your code in part (e)(i) which detects that it is invalid.

[2]

May/June 2013. P23

5 Meena wants to refine her program using the marks in `MyMarks`.

She wants the program to output:

- the word 'Failed' if the mark is less than 40,
- the word 'Distinction' if the mark is over 70,
- the phrase 'Well done' if there are more than three distinctions.

Using pseudocode, produce the algorithm that does this.

[9]





2.3.3 Selection

Oct/Nov 2013.P21

2 (b) Juan needs to design the code for a part of the program that determines where resources are kept. If the resource has:

- an ID less than 1001 it will be kept in Cabinet 1
- an ID between 1001 and 3000 it will be kept in Cabinet 2
 - even numbered IDs in Drawer 1
 - odd numbered IDs in Drawer 2
- an ID between 3001 and 5000 it will be kept in Cabinet 3

Write pseudocode that processes the variable `ResourceID` and outputs where the resource is kept. Use nested IF statements. [6]

(c) In a high-level programming language write code that processes the variable `ResourceID` and outputs where the resource is kept. Use a CASE/SELECT statement. [6]

Oct/Nov 2013.P23

2 (d) Ashvin needs to design the code for a part of the program that decides where equipment is kept.

- If the item has an ID less than 2000 it will be kept in the Physics Laboratory.
- If the item has an ID between 2001 and 4000 it will be kept in the Biology Laboratory.
- If the item has an ID between 4001 and 8000 it will be kept in the Chemistry Laboratory.
 - If the item has an ID ending in a zero it will have to be kept in a locked cabinet.

Write pseudocode that processes the variable `EquipID` and outputs where the equipment is kept, using nested IF statements. [6]

(e) In a high-level programming language write code that carries out the selections using a CASE/SELECT statement. [5]

Oct/Nov 2014.P21

3 Ahmed runs his own business. He lays floor tiles in rooms for customers. Ahmed wants a program that calculates how many tiles he needs when he inputs the measurements of the length and width of the room he is working on.

(b) The width of a room will measure at least 100 cm and less than 1000 cm.
The program must validate the width input.

Write a logic expression that is TRUE when the width stored in the variable `RoomWidth` is within the expected range. [3]

(c) Ali is going to write the program for Ahmed. The size of one floor tile is 30 cm × 30 cm. Ali knows that the following calculations are required:





2.3.3 Selection

`TilesForWidth ← RoomWidth DIV 30`

`TilesForLength ← RoomLength DIV 30`

`TilesRequired ← TilesForWidth * TilesForLength`

- (i) If the room width measures 100 cm, give the value stored in `TilesForWidth`. [1]

If the room measurements are not exact multiples of 30 cm, the number of tiles must be rounded up so that Ahmed has enough tiles.

Ali knows that he can use the MOD operator to test for this.

- (ii) Write a logic expression that is `TRUE` when the room width is not an exact multiple of 30 cm. [1]

- (iii) Ahmed wants the program to add an extra 10% to the number of tiles required in case any tiles get broken during the work.

In a high-level language, write programming statements to calculate the total number of tiles required, `TilesRequired`, after the values for `RoomLength` and `RoomWidth` have been input and validated. [5]

May/June 2015.P21/P22

3 A board game is designed for two players, O and X.

At the beginning, all cells of a 3 x 3 grid are empty.

The players take turns in placing their marker in an empty cell of the grid; player O always starts.

The game ends when one player completes a row, column or diagonal or the grid is full.

Here is one example after three turns:

		O
	O	X

- (c) Ali uses the top-down design approach for his overall program solution.

His design is as follows:

01 `GameEnd ← FALSE`

02 `CurrentPlayer ← 'O'`

03 `CALLDisplayGrid()`

04





2.3.3 Selection

```
05 REPEAT
06 CALL PlayerTakesTurn(CurrentPlayer)
07 CALL DisplayGrid()
08 IF HasPlayerWon() = TRUE
09 THEN
10 GameEnd ← TRUE
11 OUTPUT "Player", CurrentPlayer, "has won"
12 ELSE
13 IF GridFull() = TRUE
14 THEN
15 GameEnd ← TRUE
16 OUTPUT "Draw"
17 ELSE
18 CALL SwapPlayer(CurrentPlayer)
19 ENDIF
20 ENDIF
21 UNTIL GameEnd = TRUE
```

(iii) Give the line number of a statement which shows:

Selection[1]





2.3.3 Selection

May/June 2015.P23

1 (a) A high-level programming language has built-in string handling functions defined as follows:

```
MID(ThisString : STRING, x : INTEGER, n : INTEGER) RETURNS STRING  
returns a string value consisting of n characters from the string ThisString starting from  
position x.  
For example: MID("STOP", 3, 2) returns "OP"  
If the function call is not properly formed an error is generated.
```

```
CONCAT(String1 : STRING, String2 : STRING) RETURNS STRING  
returns a string value consisting of String1 followed by String2.  
For example: CONCAT("HE", "LLO") returns "HELLO"  
If the function call is not properly formed an error is generated.
```

```
LENGTH(ThisString : STRING) RETURNS INTEGER  
returns the number of characters in string ThisString.  
For example: LENGTH("Hello") returns 5  
If the function call is not properly formed an error is generated.
```

(b) Meena wants to write code to reverse a string stored in the variable `Original`. The result is to be stored in the variable `Reverse`.

(ii) A palindrome is a string that reads the same backwards as forwards.

Complete the Boolean expression to check whether the two strings, `Original` and `Reverse`, are palindromes.

```
IF .....  
THEN  
OUTPUT "This is a palindrome"  
ENDIF
```

[1]





2.3.3 Selection

Computer Science (9608)

May/June 2015.P23

2 (b) Bank customers withdraw money from their account at a cash dispenser machine using their bank card. The machine operates as follows:

∞it can dispense the following notes:

- \$50
- \$20
- \$10

∞ the maximum amount for a single withdrawal is \$500

When a customer withdraws money, they enter the amount to withdraw. (This must be a multiple of \$10).

The machine will always dispense the least possible number of notes.

A program is designed for the machine to process a withdrawal.

The following variables are used:

Identifier	Data type	Description
Amount	INTEGER	Amount to withdraw entered by the user
FiftyDollar	INTEGER	Number of \$50 notes to dispense
TwentyDollar	INTEGER	Number of \$20 notes to dispense
TenDollar	INTEGER	Number of \$10 notes to dispense
Temp	INTEGER	Used in the calculation of the number of each note required

(ii) Complete the pseudocode.

```
INPUT .....  
  
IF Amount > 500  
    THEN  
        OUTPUT "Refused - amount too large"  
    ELSE  
        .....
```



2.3.3 Selection

THEN

OUTPUT "Refused - not a multiple of \$10"

ELSE

FiftyDollar ← Amount DIV 50

Temp ←

TwentyDollar ←

Temp ←

.....

ENDIF

ENDIF

[5]

