TEMASEK JUNIOR COLLEGE

Temasek Junior College

JC H2 Computing

Problem Solving & Algorithm Design 2 – Program Flowchart

1 Introduction

Flowcharting or **block diagramming** is useful in planning the design of programs in any computer language, or for that matter, in planning the solving of any problem, regardless of whether you are using a computer.

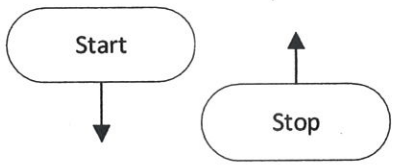
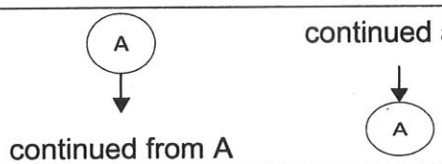
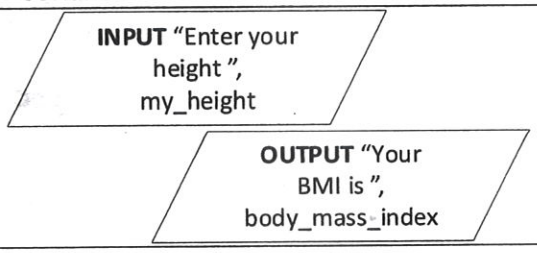
A **program flowchart** is

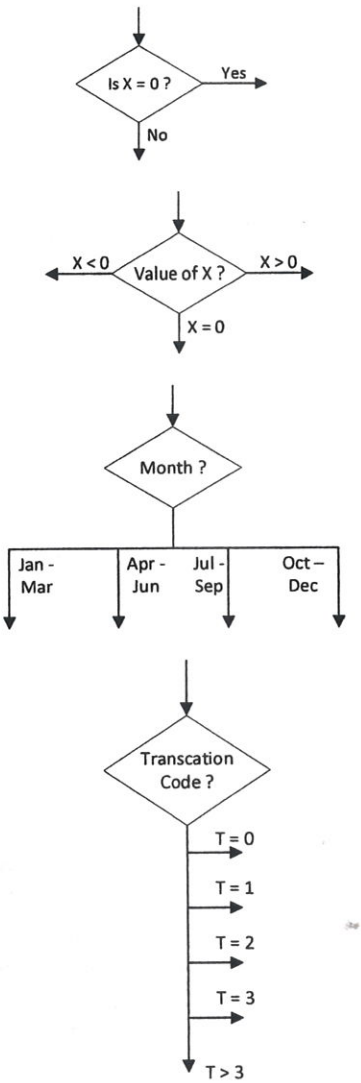
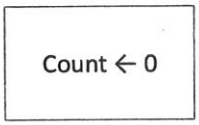
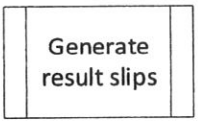
- a ***graphical representation*** of the operations involved in a computer program.
- part of the permanent record of a finished program needed for maintaining (making changes to) the program.

A program flowchart consists of:

- **program flowchart symbols** – different formalised shapes used to represent different types of program operations represented in the flowchart.
- **flow lines** – typically arrows used to link up a sequence of operations.

The table below gives the descriptions and graphical representation of common program flowchart symbols.

Types of operations	Symbols and uses	Descriptions
Start / Stop (Begin / End Terminal)		• Used to represent the starting and stopping points of a flowchart. • **Multiple stopping** points can exist but **only one starting** point.
Connector		• Used to indicate that the flowchart is continued elsewhere at an identically labeled connector point.
Input / Output		• Used for any input or output operation. • Description in the shape may include the device or medium used for the input and/or to generate the output.
Manual Input		• Indicates input operation that is performed manually.
Document		• Indicates data that can be read by people e.g. printed outputs such as receipts.

Types of operations	Symbols and uses	Descriptions
Decision		• Indicates junctures in a program where decisions are made.
Process		• Indicates any operation or sequence of instructions that does not involve any decision making such as assignment of value.
Sub-routine		• Indicates a predefined process such as the generation of result slips after every exam.

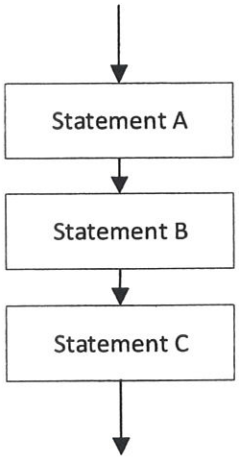
Tutorial 2.1 – Q1, 2

2. Basic Control Structures (Constructs)

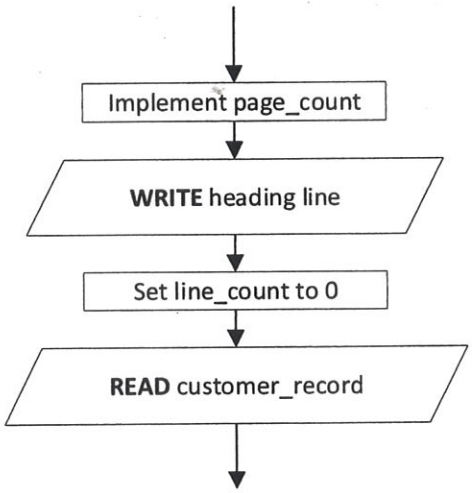
In programming, control structures are critical in determining how the program is executed. In our syllabus, we shall look at three types of basic control structures.

2.1 Sequence

A **sequence control structure** can be defined as the ordered direct (straightforward) execution of one processing step of the program after another.



Example 1
The following is an example of a sequence control structure used in the processing of customer records.



A sequence control structure can be applied in cases such as the

- intake of information,
- output of information,
- performance of arithmetic,
- assignment of values,
- ...

Many problems that we try to solve with a computer involves **inputting** data to the computer, **processing** the data and **outputting** the results in this sequence.

Example 2 – Adding three numbers
A program is required to read three numbers, add them together and print their total.

Draw a flowchart of the program.

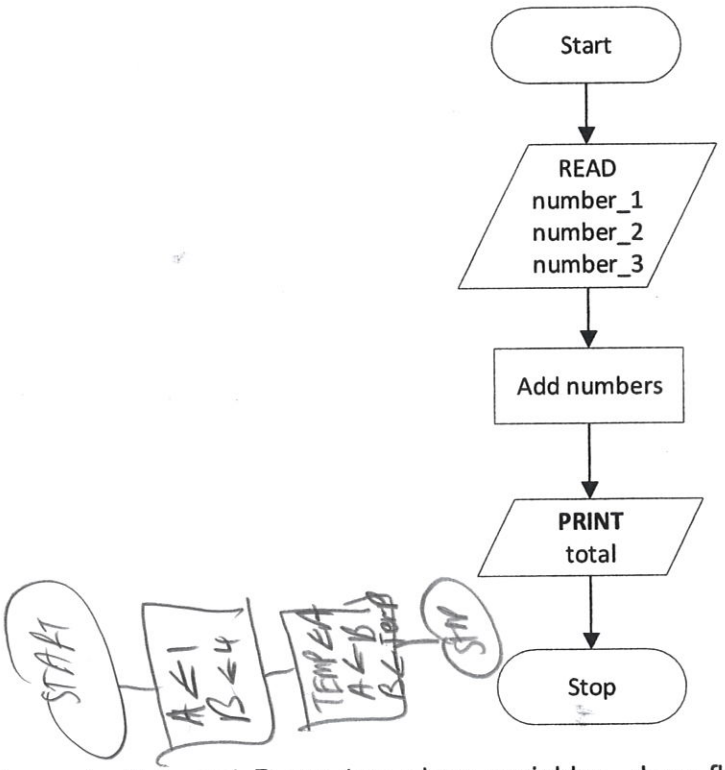
[Proposed Solution]

Problem Breakdown:
1) read three numbers,
2) add them together, and
3) print their total.

Defining Diagram:

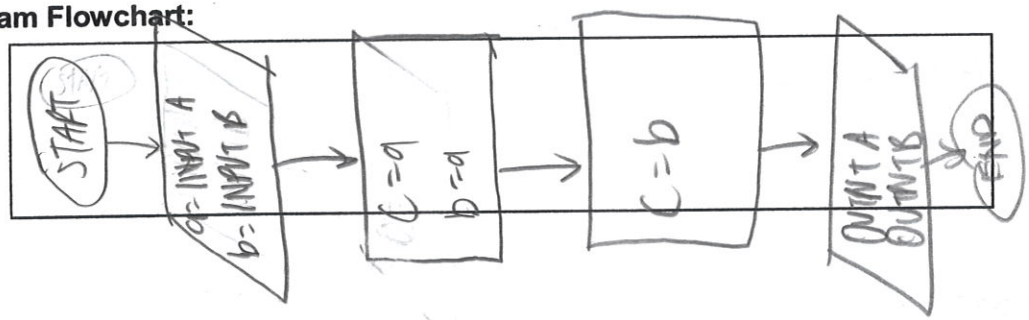
Input	Processing	Output
number_1 number_2 number_3	Read three numbers Add numbers together Print total number	Total

Program Flowchart:



Example 3 A and B are two given variables, draw flowchart for exchanging the values assigned to them.

Program Flowchart:



Example 4 – Calculating Mowing Time

A program is required to read as input the length and width of a rectangular house block, and the length and width of the rectangular house which has been built on the block. The algorithm should then compute and display the time required to cut the grass around the house, at the rate of two square meters per minute.

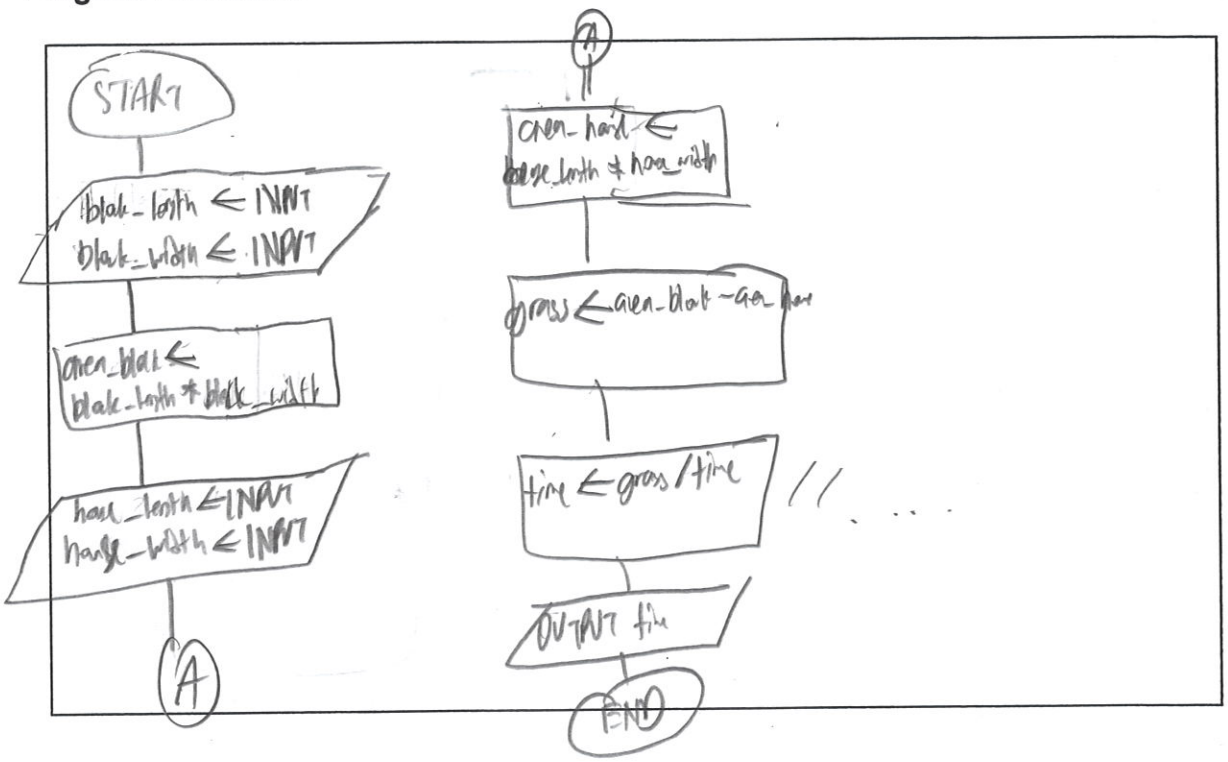
Draw a flowchart for the algorithm.

[Proposed Solution]

Defining Diagram:

Input	Processing	Output
block_length block_width house_length house_width	Prompt for block measurements Get block measurements Prompt for house measurements Get house measurements Calculate mowing area Calculate mowing time	mowing_time

Program Flowchart:

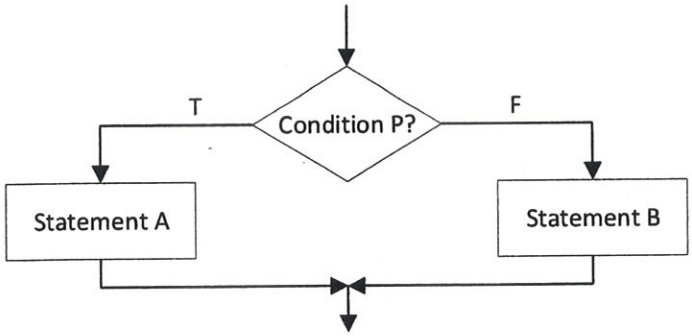


Tutorial 2.2 Q-1

2.2 Selection (Conditional Transfer)

The **selection control structure** can be defined as the presentation of a condition, and the choice between two actions depending on whether the condition is true or false.

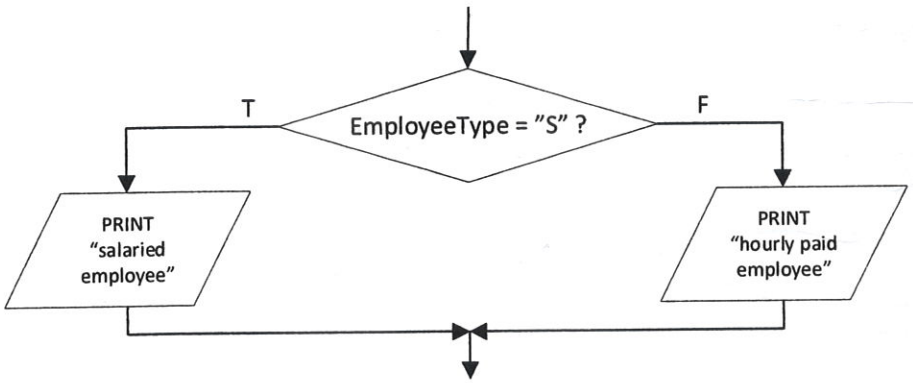
2.2.1 The diagram below shows a possible type of selection control structure.



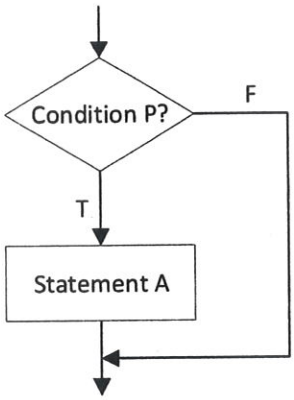
Example 5

Possible Scenario

The variable `EmployeeType` is used to indicate whether the employee is a salaried employee (S) or an hourly paid employee (H).



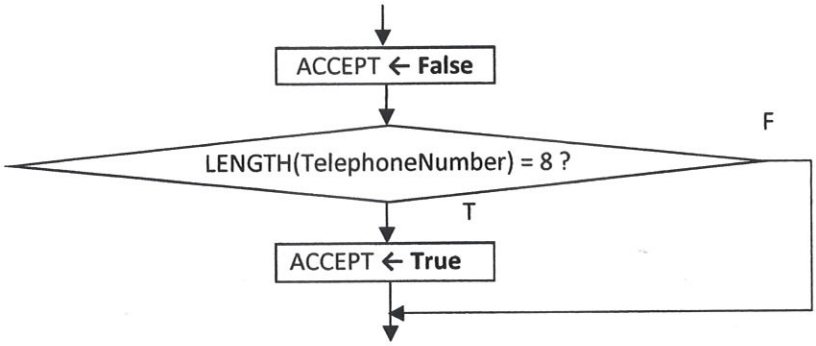
2.2.2 The diagram below shows another type of possible selection control structure.



Example 6

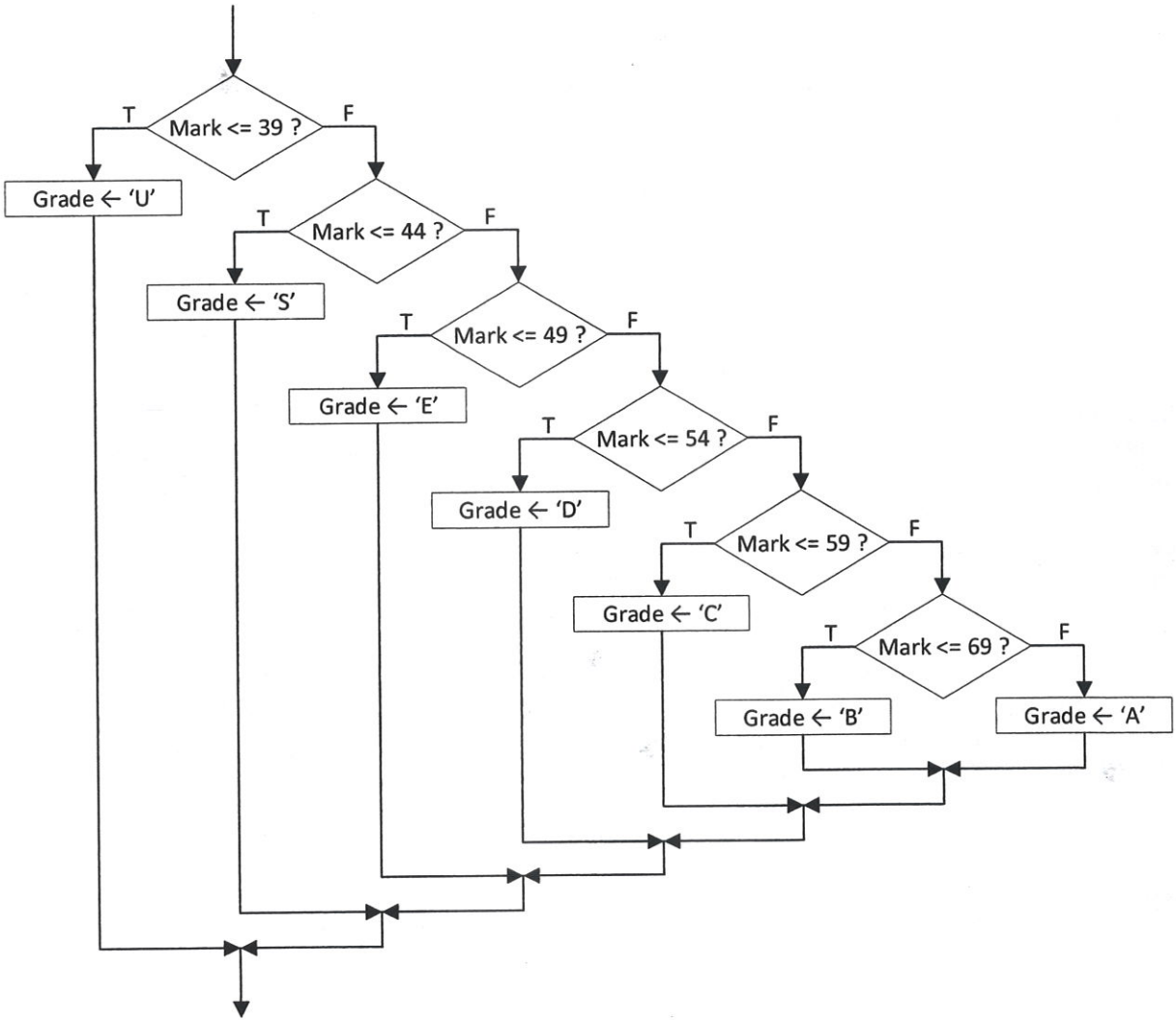
Possible Scenario

The variable `TelephoneNumber` is the telephone number of the contact and is fixed at eight digits.

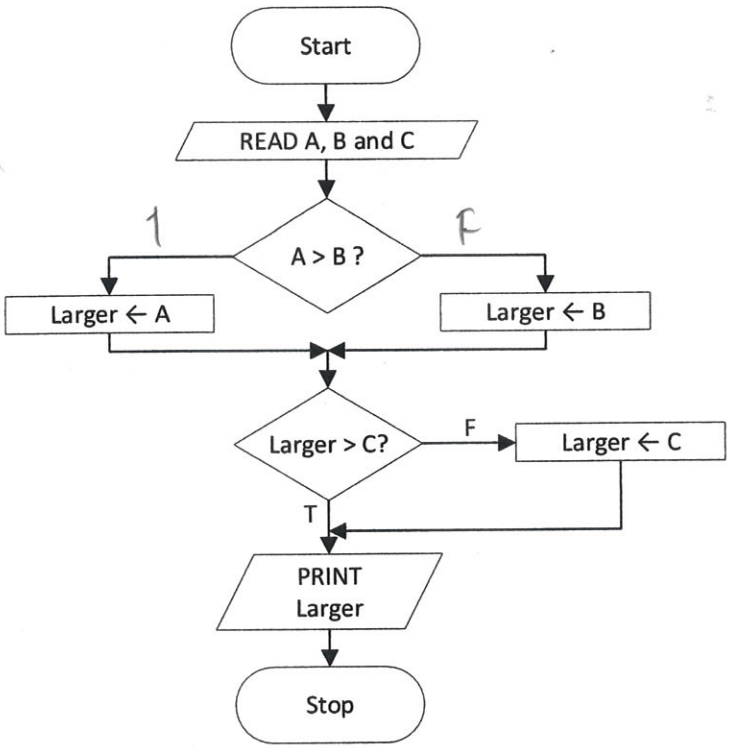


Example 7

The following shows a **nested selection control structure** for assignment of A Level grades.



Example 8
The following is the flowchart of a program used to find the largest of three numbers A, B and C.



Example 9
Design a flowchart that will receive a date in the format dd/mm/yyyy (e.g.21/01/2022) and validate it as follows:

(a) the month must be in the range 1 – 12, and
(b) the day must be in the range of 1 – 31 and acceptable for the corresponding month. (Don't forget a leap year check for February.)
[You may check for a leap year using the following steps:
Step 1: If the year is divisible by 4, go to Step 2. Otherwise, go to Step 5.
Step 2: If the year is divisible by 100, go to Step 3, Otherwise go to Step 4.
Step 3: If the year is divisible by 400, go to Step 4. Otherwise go to Step 5.
Step 4: The year is a leap year (i.e. it has 366 days).
Step 5: The year is not a leap year (i.e. it has 365 days).]

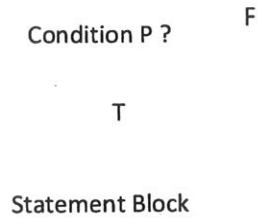
Tutorial 2.3 Q -1, 2, 4

2.3 Iteration/Repetition/Loop

The **iteration control structure** can be defined as the presentation of a set of instructions to be performed repeatedly, as long as a specified condition is true.

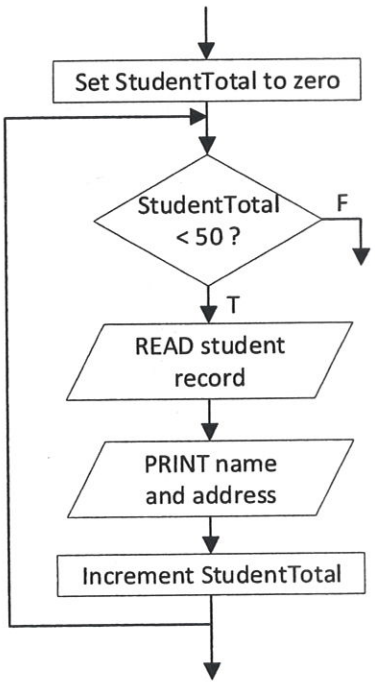
2.3.1 DO-WHILE / WHILE-END / WHILE Loop

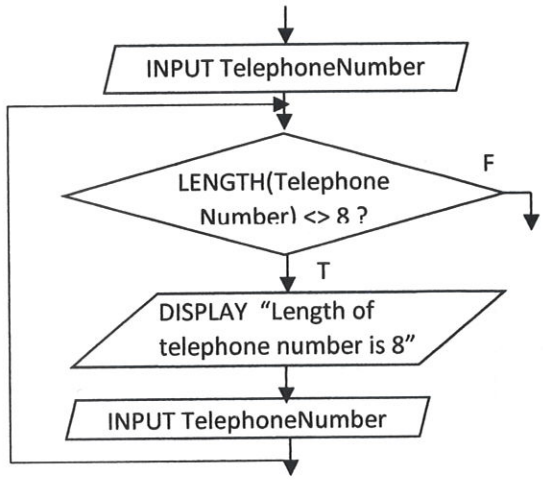
The structure of the DO-WHILE loop is as follows:



Example 10

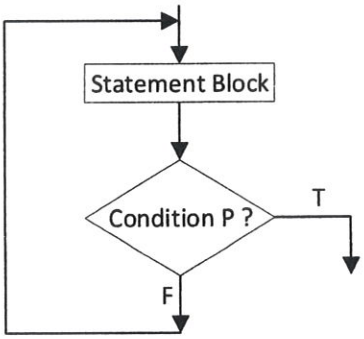
The following program flows utilises the DO-WHILE loop.





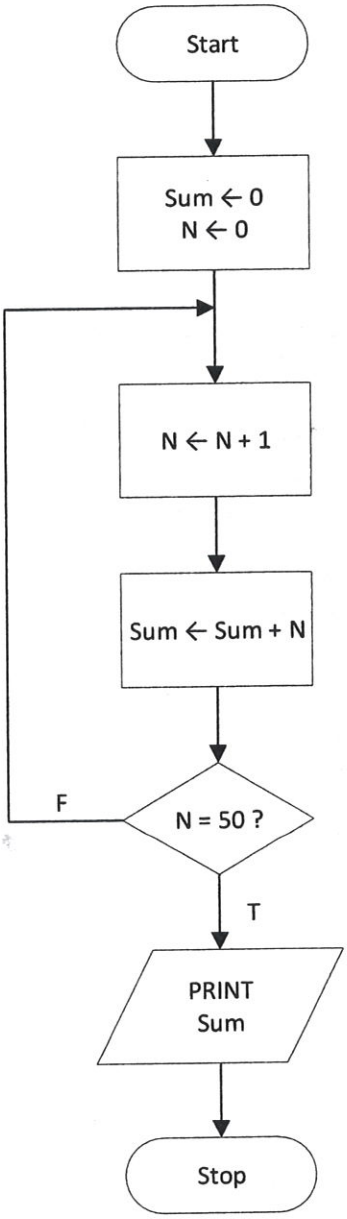
2.3.2 REPEAT-UNTIL Loop

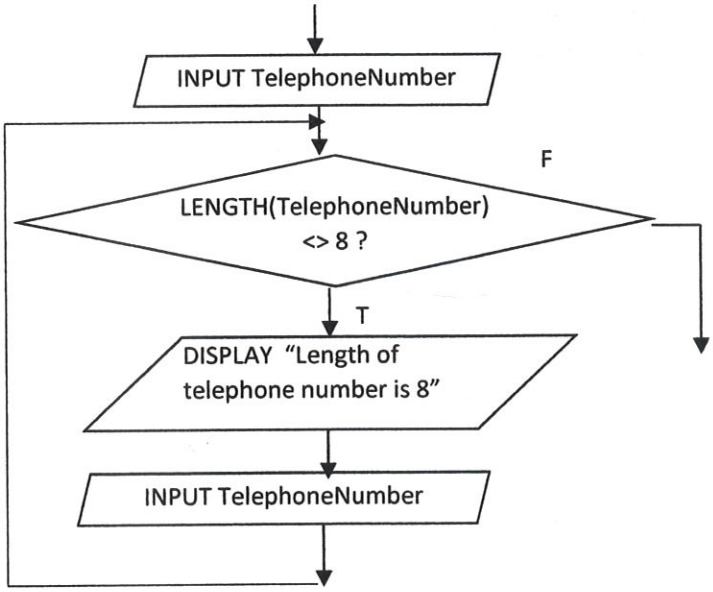
The structure of the REPEAT-UNTIL loop is as follows:



Example 11

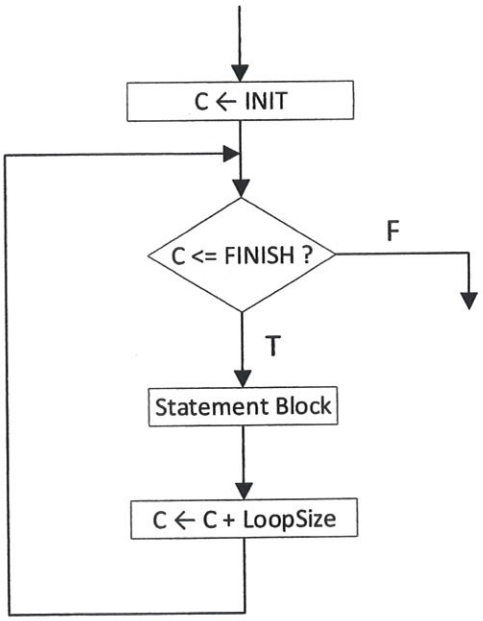
The following program flows utilises the REPEAT-UNTIL loop.





2.3.3 FOR Loop

The structure of the FOR loop is as follows:



The following lines can be used to represent a FOR loop:

FOR C = INIT **TO** FINISH **STEP** LoopSize
    Statement block
ENDFOR or NEXT or NEXT C

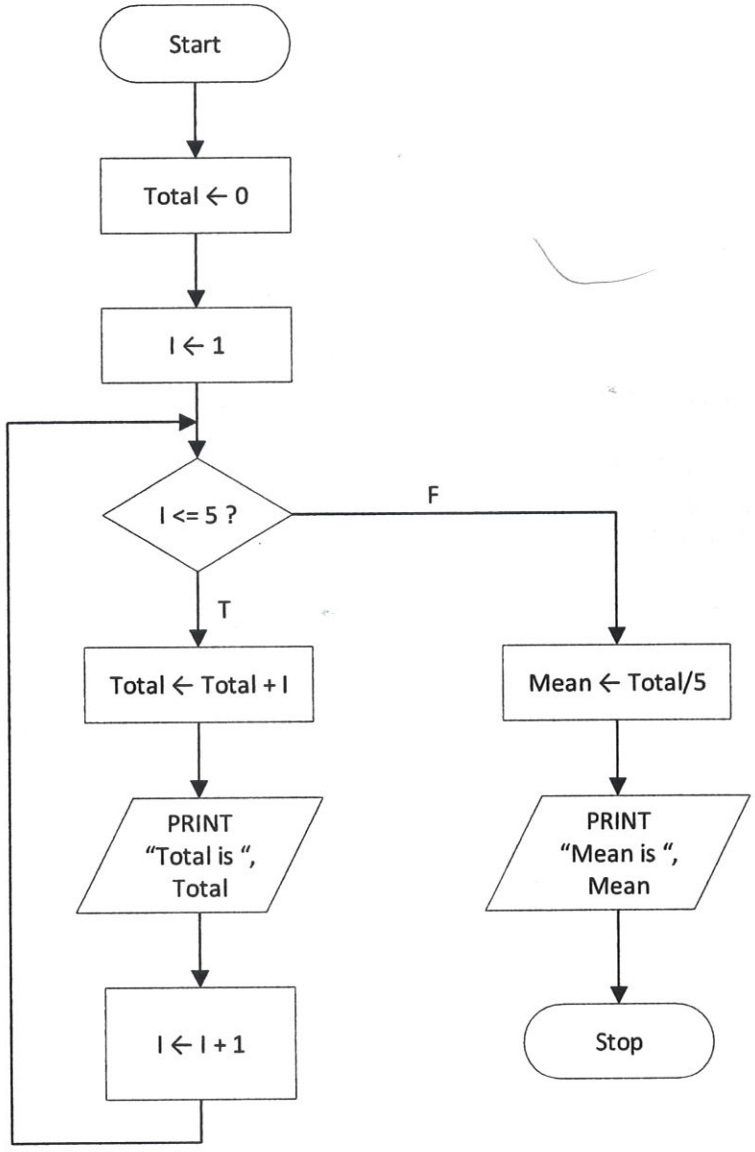
Example 12
Below is the flowchart of an algorithm that incorporates the FOR loop.

Given:

Total is integer	//stores the total of the numbers input
Mean is real	//stores the mean of the numbers input
I is integer	//stores the loop control value

Write down all the outputs in the correct order produced by the algorithm.

Total is 1
Total is 2
Total is 3
Total is 4
Total is 5
No
Mean is 1



Tutorial 2.4 Q – 1, 2, 3, 6, 9, 10

2.4 Guidelines on Flowcharting

The following are some guidelines in flow charting:

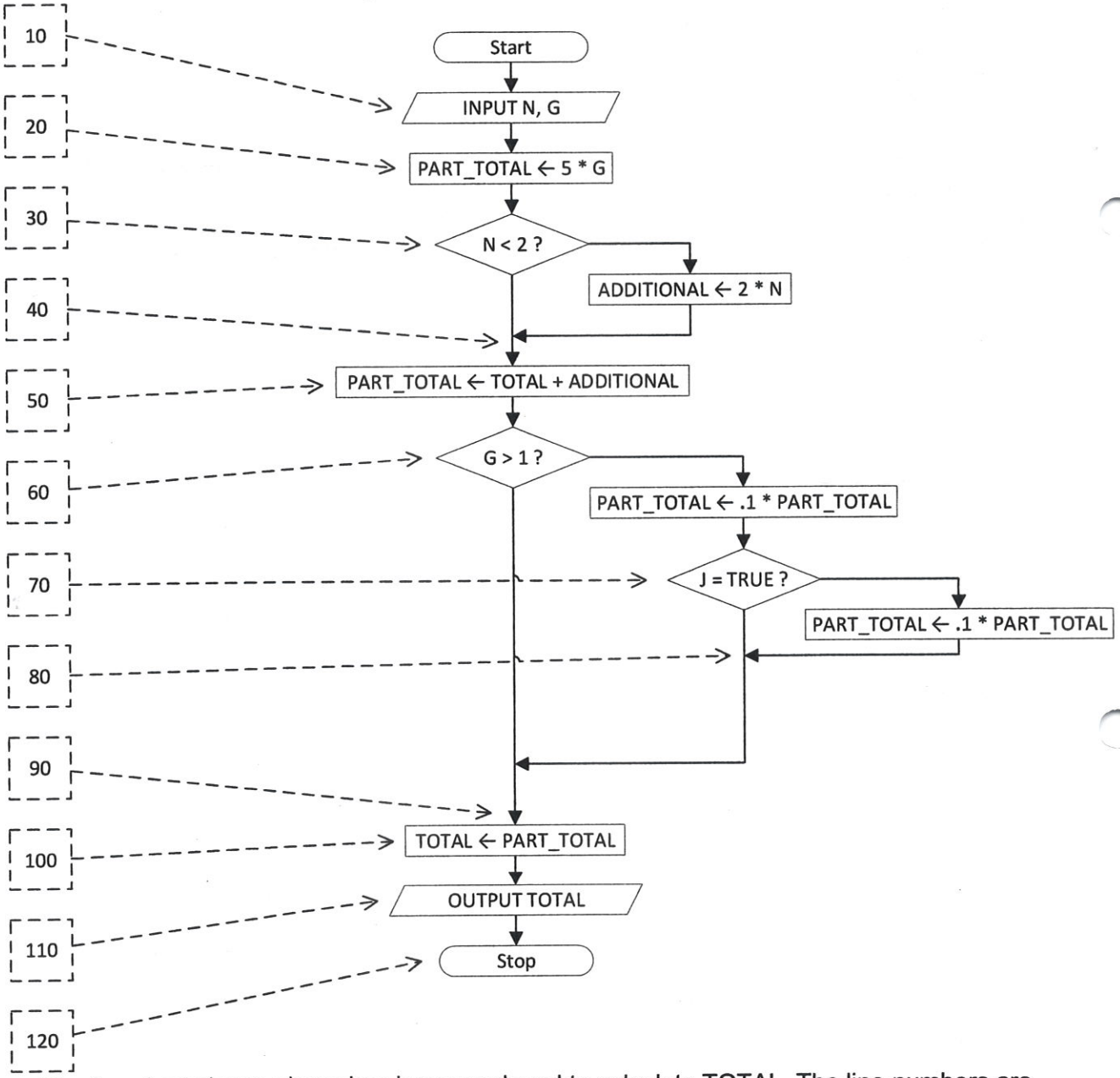
- In drawing a proper flow chart, all necessary requirements should be listed out in logical order.
- The flowchart should be clear, neat and easy to follow. There should not be any room for ambiguity in understanding the flow chart.
- The usual direction of the flow of a procedure or system is from left to right or top to bottom.
- **Only one flow line should come out from a process symbol.**
- **Only one flow line should enter a decision symbol**, but two or three flow lines, one for each possible answer, should leave the decision symbol.
- Only one flow line is used in conjunction with the terminal symbol.
- Write within standard flow chart symbols briefly. As necessary, you can use the annotation symbol to describe data or computational steps more clearly.
- If the flowchart becomes complex, it is better to use connector symbols to reduce the number of flow lines.
- Avoid the intersection of flow lines if you want to make it more effective and better way of communication.
- Ensure that the flowchart has a logical start and finish.
- It is useful to test the validity of the flowchart by passing a simple test data set through it.

Example 13

After a customer has input the necessary information to the company website, the system calculates the cost according to the following rules:

(a) $5 for each game for 1 or 2 people;
(b) an extra $2 for each person over 2 people;
(c) a 10% reduction if more than one game is booked;
(d) an additional 10% reduction if the customer is a member.

The number of people is stored in N.
The number of games is stored in G.
Whether or not the customer is a member is stored in J.
The total cost of the booking is stored in TOTAL.



The flowchart shown above has been produced to calculate TOTAL. The line numbers are to help with your answers.

(i) Explain what is happening in line 70.
(ii) There are two errors in line 30. Redraw the line as it should have been written.
(iii) Lines 50 to 90 contain a number of errors. Identify each error and state how it should be corrected so that the flowchart will output the correct TOTAL.

Example 14
Find all integers A and B between 1 and 100 such that $A^2 + B^2 = C^2$ is a perfect square.

In order to solve this problem, we will complete the following tasks:

- Analyse the problem.
- Decide on a method to use.
- Draw a flowchart.

Analysis: Before we begin our analysis, it should be noted that solutions that differ only by a permutation are to be considered identical e.g.

A = 3		A = 4
B = 4	and	B = 3
C = 5		C = 5

constitute two identical solutions.

To avoid repeating identical solutions we will seek solutions such that B > A.

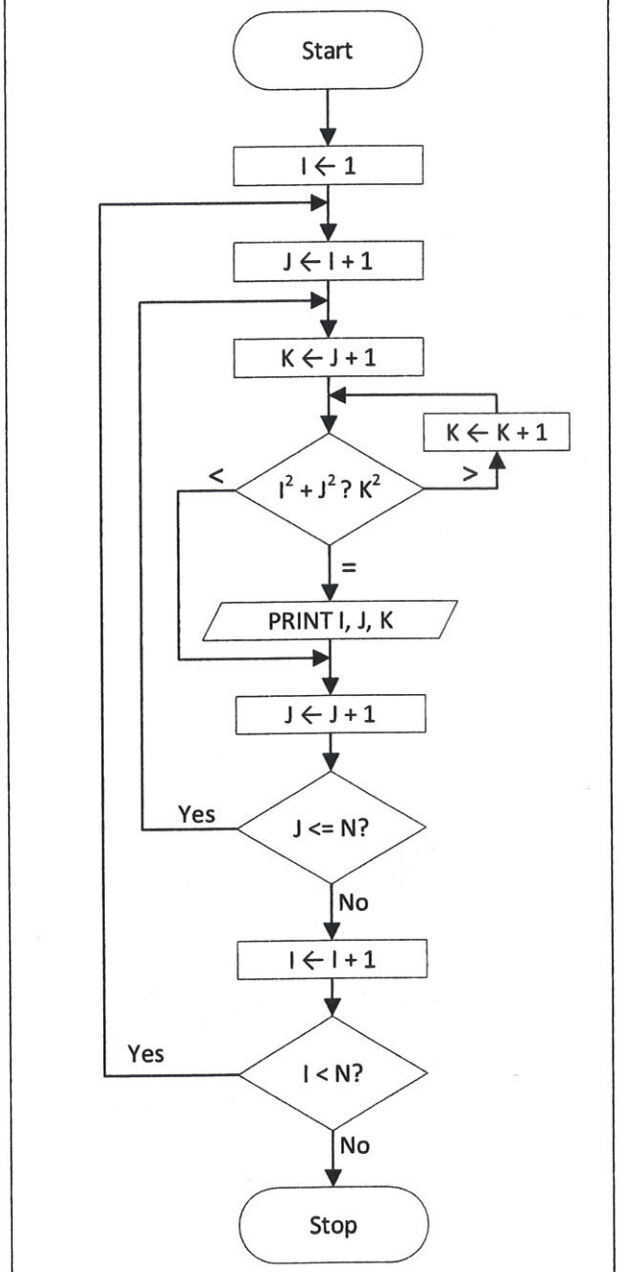
Hence let us determine if $I^2 + J^2$ is a perfect square by giving the variable I a value from 1 to 99 and the variable J a value from I + 1 to 100 (N).

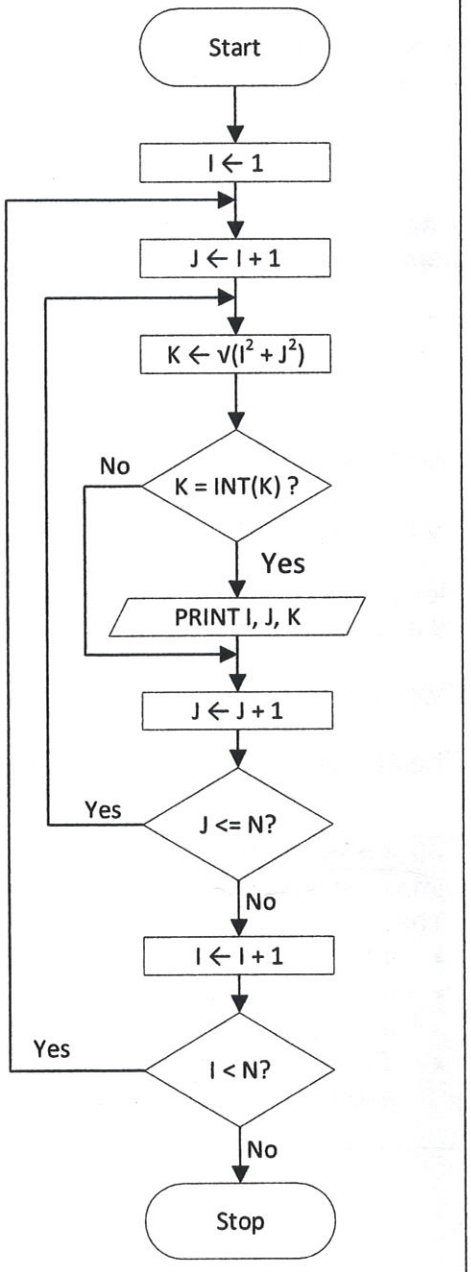
Two different approaches can be used to obtain the solution.

The approaches are shown in the next page.

First approach	**Second approach**
Implement a variable K starting from J + 1. Then, • if $I^2 + J^2 = K^2$ we have found a solution. • if $I^2 + J^2 > K^2$ increment K by one and try again. • if $I^2 + J^2 < K^2$ there is no solution for I and J.	**[Use built-in function SQRT( ) and INT( )]:** Calculate $K = \sqrt{(I^2 + J^2)}$: • if K is an integer, we have a solution. • if K is not an integer, there is no solution for I and J. • To determine whether or not K is an integer, we simply compare K with INT(K).

Solution





Tutorial 2.5 Q – 1, 2, 5

Tutorial 2

Tutorial 2.1

1* (i) What is a program flowchart?
(ii) Who produces the program flowchart in the SDLC?
(iii) Why do we use flowcharts?

2* (i) Which phase of software design do we use flowcharting?
(ii) What is the use of a program flowchart in software development?
(iii) In the maintenance phase of a software development cycle, why do we need a program flowchart?

Tutorial 2.2 [Sequence construct]

1* Draw a flowchart that will multiply two binomials. In other words, for (Ax + B)(Cx + D), you will put in data in groups of four numbers (A, B, C, D), and write out the numbers that are coefficients in the product.

Tutorial 2.3 [Selection construct]

1* Draw a flowchart for solving the quadratic equation $AX^2 + BX + C = 0$ where A is not equal to 0 for real X.

2* A home mortgage authority requires a deposit on a home loan according to the following schedule:

Loan $	Deposit
less than $25 000	5% of loan value
$25 000 – $49 999	$1250 + 10% of loan over $25 000
$50 000 – $100 000	$5000 + 25% of loan over $50 000

Loans in excess of $100 000 are not allowed.

Design an algorithm that will read a loan amount and compute and print the required deposit.

3 The tax payable on taxable incomes for employees in a certain country is set out in the following table:

Taxable income	Tax payable
From $1.00 – $4 461.99	Nil
From $4 462.00 – $17 893.99	Nil plus 30 cents for each $ in excess of $4 462.00
From $17 894.00 – $29 499.99	$4 119.00 plus 35 cents for each $ in excess of $17 894.00
From $29 500.00 – $45 787.99	$8 656.00 plus 46 cents for each $ in excess of $29 500.00
$45 788.00 and over	$11 179.00 plus 60 cents for each $ in excess of $45 788.00

Design a flowchart that will read as input the taxable income amount and calculate and print the tax payable on that amount.

4* A marathon runner records their time for a race in hours, minutes and seconds.
[Designing name of variables]

An algorithm is shown below in structured English.

INPUT race time as hours, minutes and seconds
CALCULATE race time in seconds
STORE race time in seconds
OUTPUT race time in seconds

(a) The identifier table needs to show the variables required to write a program for this algorithm.

Copy and complete the table.

Identifier	**Description**
RaceHours	The hours part of the race time.

(b) Before the program is written, the design is amended. The new design includes input of the runner's current personal best marathon time (in seconds).

The output will now also show one of the following messages:

- "Personal best time is unchanged"
- "New personal best time"
- "Equals personal best time"

(i) Show the additional variable needed for the new design.

Identifier	**Description**

(ii) Draw a flowchart for the new design.

Tutorial 2.4 [Iteration/Repetition/Loop construct]

1* Draw a flowchart that counts the number of digits in an integer
[Hint: repeatedly divided by 10].

2* Draw a flowchart to find the number of and the sum of all positive integers greater than 1000 and less than 2213 divisible by 11.

3* A man is paid 1¢ the first day, 2¢ for the second day, 4¢ the third day, and so on, doubling each day on the job for 30 days. Draw a flowchart to calculate his wages on the 30^{th} day and his total for 30 days.

4 Draw a flowchart that accepts a positive integer and reverse the order of its digits.
[DIV, MOD]

5 Draw a flowchart to sum the digits in an integer.

6* Given a number, draw flowchart to compute n factorial (written as n!) where $n \geq 0$.
$n! = n \times (n-1) \times (n-2) \times \ldots \times 3 \times 2 \times 1$

7 For a given x and a given n, draw a flowchart to compute $x^n/n!$

8 Draw a flowchart to find the amount of $100.00 deposited for one year in a saving account at 4% per year compounded four times yearly.

9* Draw a flowchart to generate the following table [**FOR-ENDFOR** loop]:

X	2 * X	X - 10	X to the power of 3	X / (-3)
1	2	-9	1	-0.333333
2	4	-8	8	-0.666667
3	6	-7	27	-1
4	8	-6	64	-1.33333
5	10	-5	125	-1.66667
6	12	-4	216	-2

10* Draw three nested **DO-WHILE/REPEAT-UNTIL/FOR-ENDFOR** loops generating the following table:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 10
```

Tutorial 2.5

1* A town contains 5000 houses. Each house owner must pay tax based on the value of the house. Houses over $200 000 pay 2% of their value in tax, houses over $100 000 pay 1.5% of their value in tax and houses over $50 000 pay 1% of their value in tax. All others pay no tax. Write an algorithm to solve this problem in the form of a flowchart.

2* The following formula is used to calculate n: n = (x * x)/(1 – x). The value x = 0 is used to stop the algorithm. The calculation is repeated using values of x until the value x = 0 is input. There is also a need to check for error conditions. The values of n and x should be output. Write an algorithm to show this repeated calculation in the form of a flowchart.

3 Write an algorithm in the form of a flowchart which takes temperatures input over a 100 day period (once per day) and outputs the number of days when the temperature was below 20°C and the number of days when the temperature was 20°C and above.

4 Write an algorithm in the form of a flowchart which:

- inputs the top speeds (in km/hr) of 5000 cars
- outputs the fastest speed and the slowest speed
- outputs the average (mean) speed of all the 5000 cars

5* A shop sells books, maps and magazines. Each item is identified by a unique 4 – digit code. All books have a code starting with 1, all maps have a code starting with 2 and all magazines have a code starting with 3. The code 9999 is used to end the algorithm.
Write an algorithm in the form of a flowchart which inputs the codes for all items in stock and outputs the number of books, number of maps and the number of magazines in stock. Include any validation checks needed.