

If-Then-Else

In general, we test a *condition*, and depending on the result, we execute some *statements*

```
If (condition) then
    statements for condition TRUE
else
    statements for condition FALSE
end if
```

Conditions

Remember that all conditions must resolve to either TRUE or FALSE values:

if (age < 16) then...

if (sum >= 12) then...

if (ratio > 4/3) then...

if (street = "Main")...

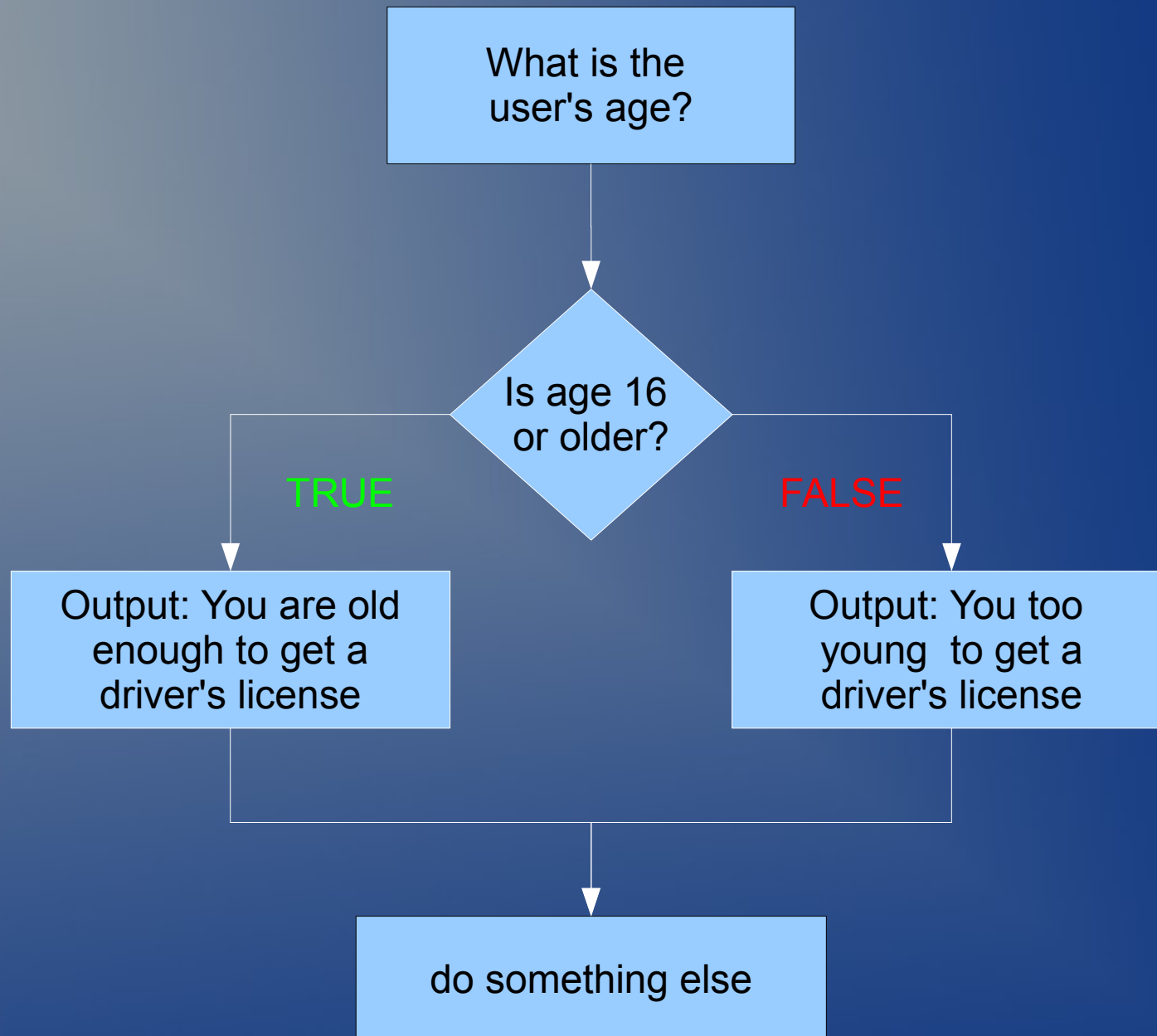
if (city not= "Ottawa")...

Flowcharts

So far, we have discussed sequential programming, where the instructions are executed one after the other.

In order to design and write programs sequentially, we need to ensure that our sequence, or flow, is correct.

It is often useful to represent our program designs using a diagram, or flowchart.



Examples

Design a solution to the following problems using a flowchart (you have already done these in Turing)

1. Ask the user their name. Give a different message depending on whether they enter your name or another name.
2. Ask the user 3 math problems and (a) tell them they are right, or (b) show them the correct answer.
3. Ask the user where they are from (e.g., province or country). Depending on their answer, give a bit of trivia on the region. Otherwise, admit you don't know anything about the area.

Testing Multiple Conditions

Not all situations can be handled by a single condition such as $(a > b)$. As our selection process becomes more complex, we can expand our use of the basic if-then-else construct.

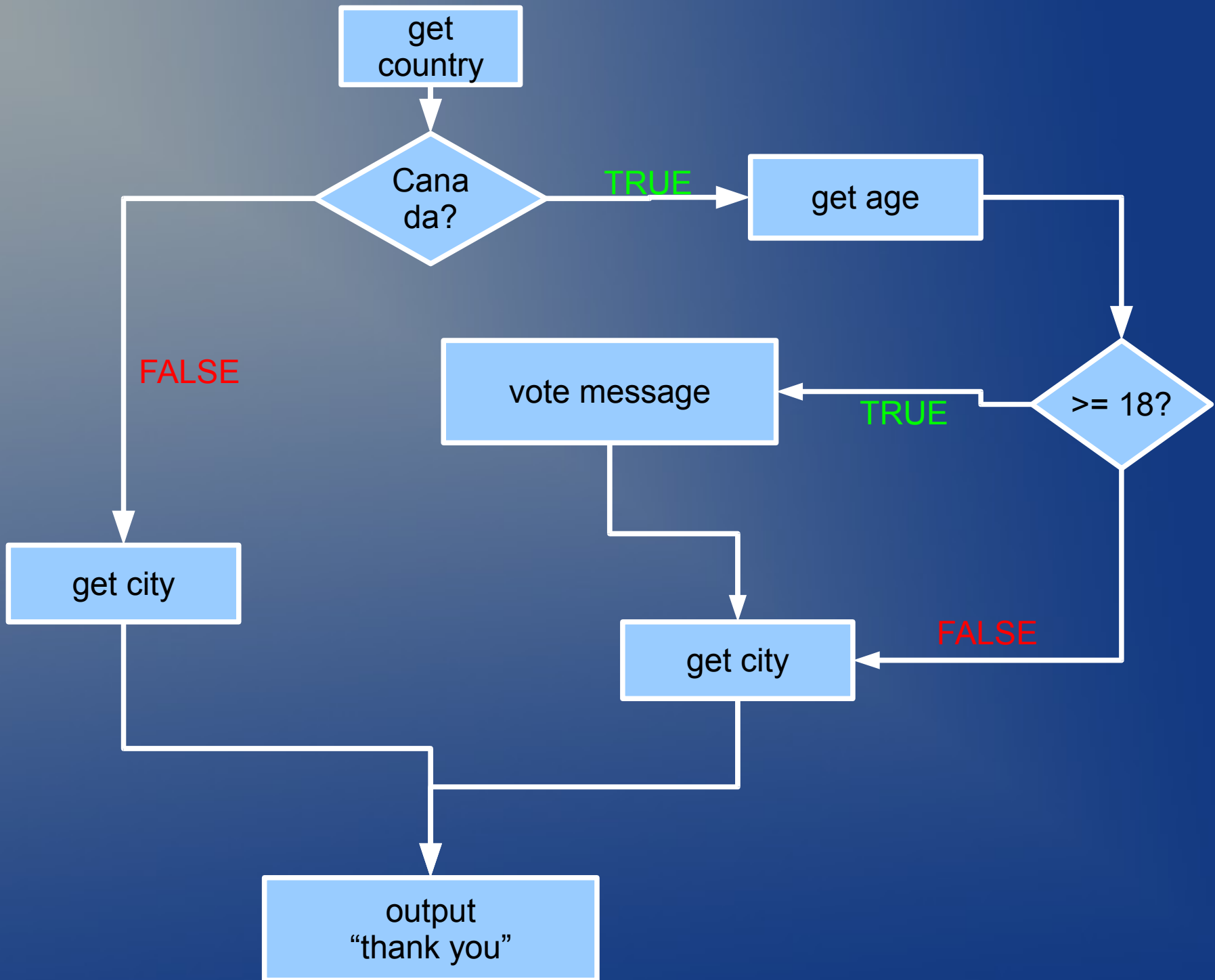
1. nested “if” statements
2. logical operators
3. “else if” statements

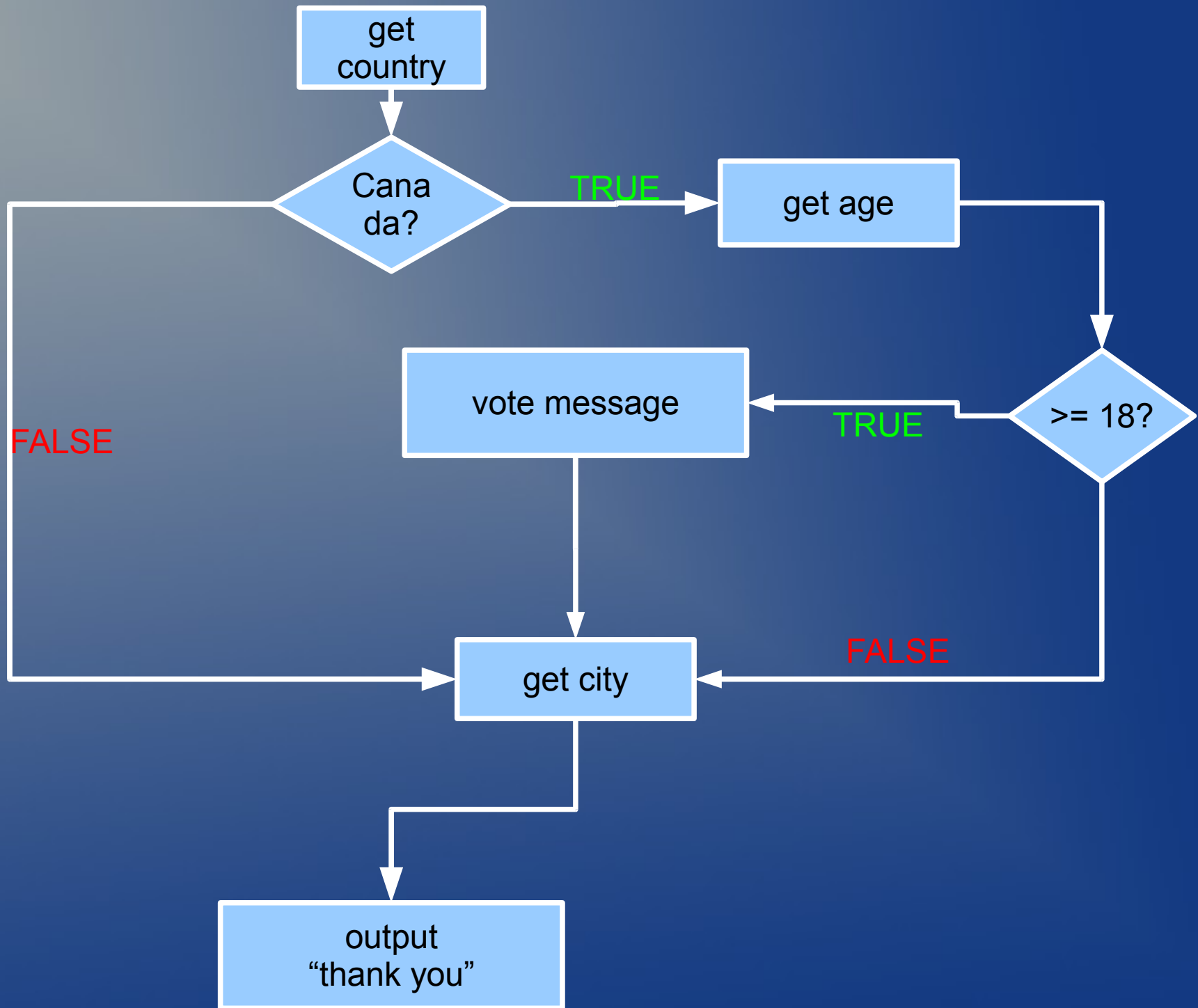
Nested “If” Statements

A “nested” statement is when we have multiple layers of the same kind of statement. In this case, multiple layers of “If” statements.

Suppose we ask someone where they live (country & city). If they are from Canada, we also want to tell them if they can vote. For any other country, we just want the city.

Try to create a flowchart for this problem.





Nested “If” - Pseudocode

input “What is your country of origin?”, country

if (country = “Canada”) then

 input “What is your age”, age

 if (age \geq 18) then

 output “Don't forget to vote!”

 end if

end if

input “What city do you live in?”, city

Logical Operators

Logical operators allow us to combine multiple conditions into a single “if” statement.

There are several logical operators, but for now we will only consider AND and OR.

AND – if all of the conditions are true, then the statement is true.

OR – if any of the conditions are true, then the statement is true.

AND Examples

If $(1 < 2)$ AND $(5 > 3)$ Then...

$(1 < 2)$ is TRUE, $(5 > 3)$ is TRUE, so the overall statement is TRUE

If $(5 = 5)$ AND ("up" = "down") Then...

$(5 = 5)$ is TRUE, but ("up" = "down") is FALSE, so the overall statement is FALSE

OR Examples

If $(1 < 2)$ OR $(5 > 3)$ Then...

$(1 < 2)$ is TRUE, $(5 > 3)$ is TRUE, so the overall statement is TRUE

If $(5 = 5)$ OR ("up" = "down") Then...

$(5 = 5)$ is TRUE, ("up" = "down") is FALSE. The OR operator only requires that at least one condition is TRUE, so the overall statement is TRUE

Else If

Sometimes we want to test several “true” conditions, and then have a single response to a “false” condition.

For example, you might the provincial capital for Ontario, Nova Scotia, and British Columbia, but none of the others.

You could program this using nested “if-then-else” statements, or with “else if” statements.

variable province is a string

input "Enter a province", province

if (province = "Ontario") then

 output "Capital = Toronto"

else

 if (province = "Nova Scotia") then

 output "Capital = Halifax"

 else

 if (province = "British Columbia") then

 output "Capital = Victoria"

 else

 output "I don't know the capital"

 end if

 end if

end if

variable province is a string

input "Enter a province", province

if (province = "Ontario") then

 output "Capital = Toronto"

else if (province = "Nova Scotia") then

 output "Capital = Halifax"

else if (province = "British Columbia") then

 output "Capital = Victoria"

else

 output "I don't know the capital"

end if

Exercises

Design at least one exercise using a flowchart

1. A program is required which will ask the user for the amount of money (positive integer only) in a bank account. It will then ask for the amount of money (positive integers) to be withdrawn.

* If the amount to be withdrawn is greater than the amount in the account, by more than \$50, the program is to display a message that the transaction is refused, and the unchanged balance is displayed.

* If the amount of money to be withdrawn is less than or equal to the amount in the account, the transaction is accepted and the new balance in the account is displayed.

* If the amount to be withdrawn is greater than the amount in the account, by up to \$50, the program is to accept the transaction and display the new balance, with a warning that the account is overdrawn.

2. Employees enter the number of hours they worked in a week. Part-time employees make \$9/hour and work up to 20 hours per week. Full-time employees make \$10/hour (over 20 hours/week). If an employee works more than 40 hours per week, they will receive overtime for the extra hours at a rate of \$15/hour.

3. A student enters 4 marks. Calculate and display their average mark. If all marks are above 80%, tell them they are on the honour roll.

4. Create a complicated problem using today's lesson and then design a solution.