

ගැටලුවක් විසඳීමට ක්‍රමලේඛයක් ලිවීම (01)

Developing a Programme to solve a problem (Programming)

Introduction to Programming Languages

(පරිගණක භාෂා හැඳින්වීම)

The purpose of a language is to allow people to communicate. Unfortunately computer at present are not intelligent enough to understand human language, because of this scientist have developed computer friendly languages. Using such a computer language you can communicate with computer. More specifically you could instruct the computer to perform a particular task.

භාෂාවක කාර්යභාරය වන්නේ මිනිසාට සන්නිවේදනය කිරීමට අවස්ථාව ලබාදීමයි. අවාසනාවකට වර්ථමානයේ ඇති පරිගණක, මානව භාෂා තේරුම් ගැනීමට තරම් බුද්ධියෙන් ප්‍රමාණවත් නොවේ. එයට හේතුව පරිගණක විද්‍යාඥයින් විසින් පරිගණකයට මිනුම් ලී භාෂා නිර්මාණය කිරීමයි. මෙවැනි භාෂාවක් භාවිතයෙන් ඔබට පරිගණකයට උපදෙස් ලබාදිය හැක.

What is a computer Language? (පරිගණක භාෂාවක් යනු කුමක්ද?)

A computer language is a set of symbols and rules used in constructing programs.

පරිගණක වැඩසටහනක් සකස්කිරීම සඳහා ඇති නීති සහ සංකේත වල එකතුවක් පරිගණක භාෂාවක් වේ.

Generation of programming languages (පරිගණක භාෂා යුග)

- 1st Generation language (1GL) (පළමු යුගයේ පරිගණක භාෂා)
- 2nd Generation languages (2GL) (දෙවන යුගයේ පරිගණක භාෂා)
- 3rd Generation languages (3GL) (තෙවන යුගයේ පරිගණක භාෂා)
- 4th Generation languages (4GL) (සිව්වන යුගයේ පරිගණක භාෂා)

1st Generation languages (1GL) (පළමු යුගයේ පරිගණක භාෂා)

- All commands are based on binary codes symbols 0 and 1 (සියලුම විධාන 0 සහ 1 යන ද්වීමය කේත ක්‍රමය මගින් පදනම් විය යුතුය යුතුය)
- Machine level programming language (පරිගණක යන්ත්‍රය මට්ටමේම භාෂාවකි)
- The machine can directly execute the machine code (යාන්ත්‍රික භාෂාව පරිගණකයට ආපූර්ව ක්‍රියාත්මක කල හැක)
- Execution is speedy because no translators are used. (පරිවර්තකයින් නොමැති නිසා යමක් සිදුවීම වේගවත්ය)
- It is very difficult to write and modify programs. (වැඩසටහනක් ලිවීම හා අලුත්කිරීම දුෂ්කර කාර්යයකි)
- It is machine dependant language (පරිගණක යන්ත්‍රය මත පදනම් වූ භාෂාවකි)
- It is a low level language (පහල පෙලේ පරිගණක භාෂාවකි)

EX: Machine Language (යාන්ත්‍රික භාෂාව)

2nd Generation languages (2GL) (දෙවන යුගයේ පරිගණක භාෂා)

- Assembly language is used (Assembly නැමති පරිගණක භාෂාව භාවිත කරන ලදී)
- Assembler is used to convert assembly language into machine language (ඇසෙම්බ්ලි භාෂාව යාන්ත්‍රික භාෂාව බවට හරවීමට ඇසෙම්බ්ලර් (Assembler) නැමති පරිවර්තකයා භාවිත කරන ලදී)

- Mnemonics and variables are used to write codes (කේත ලිවීමට Mnemonics සහ විචල්‍යන් භාවිත කරන ලදී)
- It is a machine dependant language (පරිගණක යන්ත්‍රය මත පදනම් වූ භාෂාවකි)
- Difficult to write and modify programs (වැඩසටහනක් ලිවීම හා අලුත්කිරීම දුෂ්කර කාර්යයකි)
- It is a low level language (පහල පෙලේ පරිගණක භාෂාවකි)
- Execution is fast (යමක් සිදුවන වේගය වැඩිය)

EX: Assembly Language (අැසෙම්බ්ලි භාෂාව)

3rd Generation languages (3GL) (තෙවන යුගයේ පරිගණක භාෂා)

- It is easy to understand, because it is similar to human languages (මානව භාෂාවන්ට සමාන නිසා තේරුම්ගැනීමට පහසුය)
- Compilers or interpreters are used as translators. (සම්පාදක (Compilers) සහ අර්ථවිභාසක (interpreters) නැමති පරිවර්තකයෝ භාවිත කරන ලදී)
- It is easy write and modify programs.(වැඩසටහනක් ලිවීම හා අලුත්කිරීම ඉතා පහසු කාර්යයකි)
- They are not machine dependent languages (පරිගණක යන්ත්‍රය මත පදනම් නොවූ භාෂාවවේ)
- It is a High level language (ඉහල පෙලේ පරිගණක භාෂාවකි)
- Some of the 3rd generation languages are unstructured languages such as Basic (BASIC වැනි භාෂාවන් ආකෘති රහිත ඒවා වේ)

Ex: BASIC, FORTRAN, COBOL, Pascal, Java, C++, Visual Basic, Python

4th Generation languages (4GL) (සිව්වන යුගයේ පරිගණක භාෂා) (Artificial Language / කෘතීම බුද්ධිය සහිත භාෂා)

- Artificial Intelligence is used in these languages (මෙම පරිගණක භාෂා කෘතීම බුද්ධිය භාවිතකරයි)
- It consumes less time to write a program (වැඩසටහනක් ලිවීමට අඩු කාලයක් වැයවේ)
- Very easy to write and modify .(වැඩසටහනක් ලිවීම හා අලුත්කිරීම ඉතා පහසු කාර්යයකි)
- It is a High level language (ඉහල පෙලේ පරිගණක භාෂාවකි)

EX: Prolog, LISP

Classification of Programming Languages (පරිගණකභාෂා වර්ගීකරණය)

1 Low Level Languages (පහල පෙලේ පරිගණක භාෂා)

First and second generation languages are considered low level languages. (පළමු සහ දෙවන යුගයේ භාෂා මෙම වර්ගයට අයත්වේ)

EX: Machine Language (යාන්ත්‍රික භාෂාව)

Assembly Language (අැසෙම්බ්ලි භාෂාව)

2 High Level Languages (ඉහල පෙලේ පරිගණක භාෂා)

- 3GL and 4GL are considered high level languages (තෙවන සහ සිව්වන යුගයේ භාෂා මෙම වර්ගයට අයත්වේ)

EX: BASIC, FORTRAN, COBOL, Pascal, Java, FOCUS, IDEAL, dbase III plus, Visual basic, C++,

Small Talk, Visual.net, Prolog, LISP

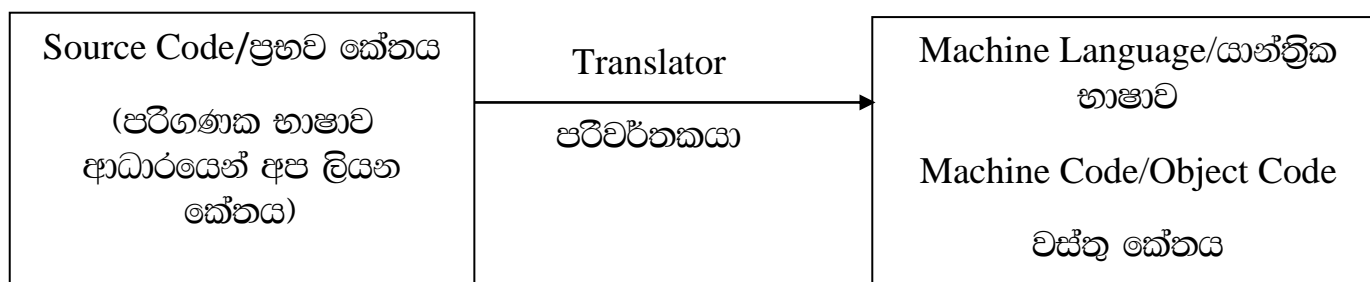
Language Translator (භාෂා පරිවර්තකයා)

Translators are used to convert source code into machine (object) code. These are three types of translators

ප්‍රභව කේතය (අප විසින් පරිගණක භාෂාවක් ආධාරයෙන් ලියන කේතය (Source Code) යාන්ත්‍රික භාෂාව (Machine Language) බවට හැරවීමට පරිවර්තකයින් භාවිත කරනු ලබයි.

පරිවර්තකයින් ප්‍රධාන වර්ග තුනක් වේ

1. Assembler (ඇසෙම්බ්ලර්)
2. Compiler (සම්පාදක)
3. Interpreter (අර්ථවිභ්‍යාසක)



Assembler (ඇසෙම්බ්ලර්)

- It is a low-level language Translator. (පහල පෙලේ පරිවර්තකයෙකි)
- It is a software program that converts assembly language into machine language. (ඇසෙම්බ්ලි භාෂාව යාන්ත්‍රික භාෂාව බවට හැරවීමට මෙම වැඩසටහන භාවිත කරන ලදී)
- Converts Mnemonics into machine code (Mnemonics යාන්ත්‍රික භාෂාවට හරවයි)

EX: Assembly language (ඇසෙම්බ්ලි භාෂාව)

Compiler (සම්පාදක)

- It is a translator of high level languages (ඉහල පෙලේ පරිවර්තකයෙකි)
- Converts a whole program into machine language at once (ප්‍රභව කේතය (අප විසින් ලියන කේතය) යාන්ත්‍රික භාෂාව බවට එකවර පරිවර්තනය කරයි)

EX: Pascal, Visual Basic

Interpreter (අර්ථවිභ්‍යාසක)

- It is a translator of high level languages (ඉහල පෙලේ පරිවර්තකයෙකි)
- Translate source code into machine code line by line (ප්‍රභව කේතය (අප විසින් ලියන කේතය) යාන්ත්‍රික භාෂාව බවට පේලියෙන් පේලිය පරිවර්තනය කරයි/වරකට එක් පේලියක් බැගින් පරිවර්තනය කරයි)

EX: Python

Solving Problems (ගැටළු විසඳීම)

Steps of solving a problem (ගැටළුවක් විසඳීමේ පියවර)

- 1 Understanding the problem/Analysing the problem
(ගැටළුව හඳුනාගැනීම/ගැටළුව විශ්ලේෂණය)
- 2 Developing an Algorithm for the problem
(ගැටළුව සඳහා ඇල්ගොරිතමයක් ගොඩනැගීම)

- 3 Cording Programme (පරිගණක භාෂාවක් ආධාරයෙන් කේත කිරීම)
- 4 Testing and debugging (පරීක්ෂාවට ලක්කිරීම හා වැරදි නිවැරදි කිරීම)
- 5 Execute the programme on the input data (වැඩසටහන ආදාන දත්තමත ක්‍රියාත්මක කිරීම)

(01) ගැටළුව හඳුනාගැනීම/ගැටළුව විශ්ලේෂණය (Understanding the problem/Analysing the problem)

ගැටළුවක් විසඳීම සඳහා යොදාගන්නා අමුද්‍රව්‍ය ආදාන (Input) ලෙස හැඳින්වේ. එය විසඳීමෙන් ලබා ගන්නා ප්‍රතිඵලය, ප්‍රතිදානය (Output) ලෙස හැඳින්වේ. ආදාන, ප්‍රතිදාන බවට පත්කිරීම, ක්‍රියාවලිය හෙවත් සැකසීම (Process) ලෙස හැඳින්වේ. ක්‍රියාවලියක් පියවරෙන් පියවර සිදු වන අතර ඒවා අනුපිලිවෙලකින් දැක්වීම ඉතා වැදගත්වේ. ගැටළුවක් විශ්ලේෂණය කිරීමේ දී එහි ආදාන, ක්‍රියාවලිය හා ප්‍රතිදාන වෙන් වෙන් ව හඳුනා ගනු ලබයි.

The raw materials that are used to solve a problem are known as the 'input'. The result obtained after solving a problem is known as the 'output'. Converting input to output is called the 'process'. A process takes place step by step and it is very important to understand the order of process. When analyzing a problem, the input, processing and output are identified separately.

ගැටළුව : තැපැල් කිරීමට සුදුසු ලිපියක් සෑදීම

ආදාන : ලිපිය ලිවීමට සුදුසු කොළයක්, පෑනක්, ලියුම් කවරයක්, මුද්දර සහ මැලියම්

- ක්‍රියාවලිය** :
1. ලිපිය ලිවීම
 2. ලිපිය නවා ලියුම් කවරයට බහා ලීම
 3. ලියුම් කවරය ඇලවීම
 4. යා යුතු ලිපිනය ලියුම් කවරයේ ලිවීම
 5. මුද්දර ඇලවීම

ප්‍රතිදානය : තැපැල් කිරීමට සුදුසු තත්ත්වයේ පවතින ලිපියක්

සටහන : මෙම ක්‍රියාවලියේ පියවර අංක 4 සහ 5 හුවමාරු කර සිදු කළ හැකි ය. එහෙත් අනෙකුත් පියවර අනුපිලිවෙලින් ම සිදු කළ යුතු වේ.

Problem : Preparing a letter which can be posted.

Input : A sheet of paper, suitable to write the letter on, a pen, an envelope, a stamp and Glue

- Process**
1. Writing the letter
 2. Folding the letter and putting it into the envelope
 3. Pasting the envelope
 4. Writing the recipient's address on the envelope
 5. Sticking the stamp

Output : A letter ready to be posted.

Note : Steps No. 4 and 5 in this process can be interchanged. However, the other steps should be followed in the order indicated.

ගැටලුව : සංඛ්‍යාවක් ඔත්තේ ද ඉරට්ටේ ද යන්න සෙවීම

ආදානය : සංඛ්‍යාව

ක්‍රියාවලිය : සංඛ්‍යාව දෙකෙන් බෙදා ශේෂය (ඉතිරිය) සෙවීම

ශේෂය = 0 නම් ඉරට්ටේ සංඛ්‍යාවක් බව තීරණය කිරීම

ශේෂය = 1 නම් ඔත්තේ සංඛ්‍යාවක් බව තීරණය කිරීම

ප්‍රතිදානය : සංඛ්‍යාව ඔත්තේ හෝ ඉරට්ටේ බව දැක්වීම

Problem : Finding whether a number is odd or even

Input : Number

Process : Dividing the number by 2

Deciding that the number is even if the remainder = 0

Deciding that the number is odd if the remainder = 1

Output : Indicating whether the number is odd or even

විකල්ප විසඳුම් හඳුනාගැනීම (Identifying the alternative solutions?)

ගැටලුවක් සඳහා විසඳුම් එකකට වඩා වැඩියෙන් පවතී නම් ඒවා විකල්ප විසඳුම් ලෙස හැඳින්වේ. මෙය ගැටලුවේ ස්වභාවය මත තීරණය වන්නකි.

If there is more than one solution to a given problem such solutions are called alternative solutions. Such solutions depend on the nature of the problem.

නිදසුන 1

ඔබ පාසලට පැමිණෙන්නේ පාසල් බසයකින් යැයි සිතන්න. පාසලට පැමිණෙමින් සිටින විට බස් රථයේ දෝෂයක් නිසා අතරමග නැවතුණි නම් ඔබට පාසල වෙත පැමිණීමට හැකි වෙනත් ක්‍රම ගැන සිතනවා නේ ද? මෙසේ ඔබ සිතන්නේ පාසල වෙත යාමට හැකි වෙනත් විකල්ප විසඳුම් ය.

1. පාසලට එන වෙනත් පාසල් බස් රථයකින් පාසල වෙත පැමිණීම
2. ඔබ ළඟ මුදල් ඇති නම් ලංගම හෝ පෞද්ගලික බස් රථයකින් පාසල වෙත පැමිණීම
3. පාර දිගේ පයින් ම ගමන් කර පාසල වෙත පැමිණීම
4. කෙටි මාර්ගයකින් පයින් ගමන් කර පාසල වෙත පැමිණීම
5. දෙමව්පියන්ට මේ බව දැන්විය හැකි නම් ඔවුන්ගේ සහාය ඇති ව පාසල වෙත පැමිණීම
6. විශ්වාසවන්ත අයෙකුගේ සහාය ඇති ව මෝටර් රථයකින් හෝ මෝටර් සයිකලයකින් පාසල වෙත පැමිණීම

එදින අනිවාර්යයෙන් ම පාසලට පැමිණීමට අවශ්‍ය දිනයක් නම් මෙම විකල්ප විසඳුම් අතරින් හොඳ විසඳුමක් තෝරා ගතයුතුය. ඒ අනුව යම් ගැටලුවකට විසඳුම් සමූහයක් (කුලකයක්) පවතී නම් ඒ පිළිබඳ ව සලකා බලා හොඳ විසඳුමක් තෝරා ගැනීම වඩාත් උචිත වේ. මෙසේ ගැටලුවකට පවතින සියලු විසඳුම්, විසඳුම් අවකාශය (Solution Space) ලෙස හැඳින්වේ. පරිගණක ක්‍රම ලේඛනයේ දී ද විවිධ විසඳුම් හඳුනාගෙන හොඳ ම විසඳුම තෝරා ගැනීම කළ යුතු ය. එවිට සරල කෙටි ක්‍රමලේඛනයක් නිර්මාණය කිරීමට හැකි වේ.

All the solutions pertaining to a problem are called solution space. In computer programming also, various solutions should be identified and an appropriate solution should be selected. Then we can create a short, simple programme.

නිදසුන 2

සාප්‍රකෝණාස්‍රයක පරිමිතිය සෙවීමට ඇති විසඳුම් අවකාශය සොයා බලමු. මෙම ගැටලුවට අදාළ ආදාන, ප්‍රතිදාන සහ සැකසීම විශ්ලේෂණය කොට දක්වමු.

- ආදාන : සාප්‍රකෝණාස්‍රයේ දිග හා පළල
- සැකසීම : පරිමිතිය ගණනය කිරීම
- ප්‍රතිදාන : පරිමිතිය

පරිමිතිය ගණනය කිරීම සඳහා විසඳුම් අවකාශය සොයමු.

- 1 වන විසඳුම. පරිමිතිය = දිග + පළල + දිග + පළල
- 2 වන විසඳුම. පරිමිතිය = (දිග x 2) + (පළල x 2)
- 3 වන විසඳුම. පරිමිතිය = (දිග + පළල) x 2

මෙම විසඳුම් අතරින් එකතු කිරීම පිළිබඳ ව පමණක් අවබෝධයක් ඇති අයෙකුට හොඳ ම විසඳුම ලෙස 1 වන විසඳුම තෝරා ගත හැකි ය. ගුණ කිරීම හා එකතු කිරීම පිළිබඳ ව අවබෝධයක් ඇති අයෙකුට හොඳ ම විසඳුම ලෙස 3 වන විසඳුම තෝරා ගත හැකි ය. එයට හේතුව එකතු කිරීමේ හා ගුණ කිරීමේ කාරක අවම ප්‍රමාණයක් පවතින බැවිනි.

නිදසුන් 3

ශිෂ්‍යයකු තොරතුරු හා සන්නිවේදන තාක්ෂණ විෂයය සඳහා ලබා ගත් ලකුණු සංඛ්‍යාව 350 අඩු නම් ඔහු අසමත් බව හෝ 35 හෝ ඊට වැඩි නම් ඔහු සමත් බව ද දැක්වීම

- ආදාන : ලකුණු සංඛ්‍යාව
- සැකසීම : ලබා ඇති ලකුණු ප්‍රමාණය 35 සමග සැසඳීම

විසඳුම 1. ලකුණු 350 අඩු නම්
ප්‍රතිඵලය = අසමත්
එසේ නොවේ නම්
ප්‍රතිඵලය = සමත්

විසඳුම 2. ලකුණු 35 හෝ ඊට වඩා වැඩි නම්
ප්‍රතිඵලය = සමත්
එසේ නොවේ නම්
ප්‍රතිඵලය = අසමත්

ප්‍රතිදාන : සමත් හෝ අසමත් බව

(02) ගැටළුව සඳහා ඇල්ගොරිතමයක් ගොඩනැගීම (Developing an Algorithm for the problem)

What is an Algorithm? (ඇල්ගොරිතමයක් යනු කුමක්ද?)

An algorithm is a step-by-step procedure for solving a problem. (An algorithm is a method to show the steps in solving a problem)

(ගැටළුවක් විසඳීම සඳහා අනුගමනය කලයුතු පියවර සියල්ල අනුපිලිවෙලින් දැක්වූ ක්‍රමවේදයක් ඇල්ගොරිතමයක් වේ (ගැටළුවක් විසඳීම පියවරෙන් පියවර දැක්වූ ක්‍රමයක්))

උදාහරණ - ලිපියක් තැපැල් කිරීම සඳහා ඇල්ගොරිතමය (Developing an algorithm to post a letter.)

- | | |
|-------------------------|------------------------------------|
| (1) ලිපිය ලිවීම | Write the letter |
| (2) ලිපිය නැවීම | Folding the letter |
| (3) ලියුම් කවරයකට දැමීම | Inserting the letter in an envelop |
| (4) ලිපිනය ලිවීම | Writing the address |
| (5) මුද්දර ඇලවීම | Sticking the stamp |
| (6) තැපැල් කිරීම | Posting the letter |

ඉහත පියවර ලිපියක් තැපැල් කිරීම සඳහා වූ ඇල්ගොරිතමය වේ. මෙම ඇල්ගොරිතමයේ (1), (2) සහ (3) වන පියවර අනුපිලිවෙලින් ම සිදුකළ යුතු ය. (4) සහ (5) පියවර මාරුකළ හැකි ය. එයට හේතුව ලිපිනය ලියා මුද්දර ඇලවිය හැකි ය. එමෙන් ම මුද්දරය ඇලවා ලිපිනය ලිවිය හැකි ය. ඒ අනුව ඇල්ගොරිතමයක අනුපිලිවෙලින් ම ක්‍රියාත්මක කළ යුතු පියවර පවතින අතර සමහර විට පියවර කිහිපයක අනුපිලිවෙළ වෙනස් කළ ද එමගින් ඉදිරිපත් කෙරෙන ක්‍රියාවලියට හානියක් සිදු නොවේ.

Representation of Algorithm (ඇල්ගොරිතම නිරූපනය)

- 1 Graphical Representation (රූපමය නිරූපනය)
- 2 Textual Representation (ලිඛිතමය නිරූපනය)


Flow Charts (ගැලීම් සටහන්)


A flow chart is a graphical representation of an algorithm. Each step in the flow chart is represented by a designed symbol and is linked with arrows showing the direction of data flow. A standard set of symbols has been introduced to be used in flow chart.


(ඇල්ගොරිතමයක විධානයන් ක්‍රියාකලයුතු ආකාරය අනුපිලිවෙලින් රූපමය සංකේත මගින් දැක්වූ සටහනක් ගැලීම්සටහනක් ලෙස හඳුන්වනු ලබයි. ගැලීම් සටහනක් නිර්මාණයේදී භාවිත කල යුතු සම්මත සංකේත හඳුන්වාදී ඇත.)

Symbols Used in Flow chart

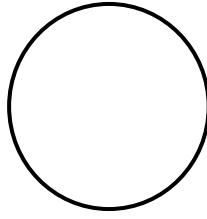
ගැලීම් සටහනකදී භාවිතයට ගන්නා සංකේත

<p>1 Start/Stop, Begin/End ආරම්භය/අවසානය Terminator</p>	
---	--

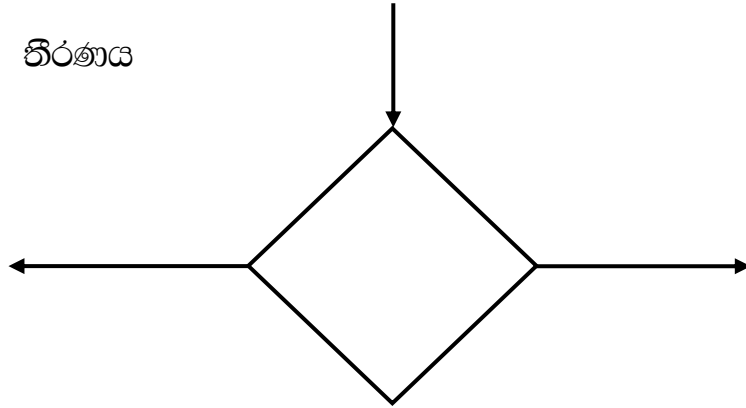
<p>2 Input/Output ආදාන/ප්‍රතිදාන</p>	
--------------------------------------	--

<p>3 Process සැකසීම/ක්‍රියාවලිය</p>	
-------------------------------------	--

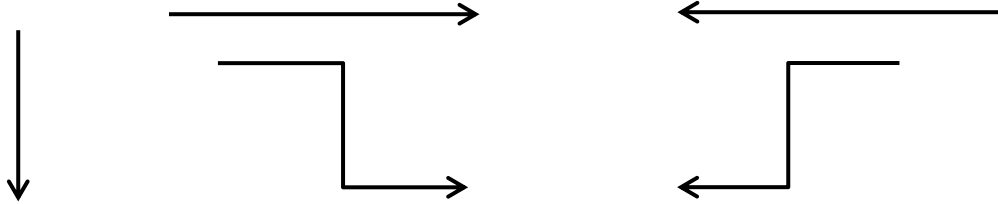
4 Connector සම්බන්ධකය



5 Decision තීරණය



6 Flow Lines/Flow direction ගැලිම් රේඛා/ගැලිම් දිශාව



ප්‍රශ්න අංක 01

Draw a flow chart to add two numbers given by the user and display the sum (total) of those two numbers
(භාවිතකරන්නා විසින් ලබාදෙන සංඛ්‍යා දෙකක එකතුව සොයා එම එකතුව පෙන්වන ගැලිම් සටහනක් අඳින්න)