

1. A selection control structure is a programming structure used to make a decision based on one or more conditions./ තේරීම් පාලන ව්‍යුහයක් යනු කොන්දේසි එකක් හෝ කිහිපයක් මත පදනම්ව තීරණයක් ගැනීමට භාවිතා කරන ක්‍රමලේඛන ව්‍යුහයකි.
2. Simple selection has one condition with two possible options. If the condition is true, one option is selected; if it is false, the other option is selected./ සරල තේරීමකදී ඇත්තේ එක් කොන්දේසියක් සහ ඒ ආශ්‍රිත විකල්ප දෙකකි. කොන්දේසිය සත්‍ය නම් එක් විකල්පයක් තෝරාගනු ලබන අතර, එය අසත්‍ය නම් අනෙක් විකල්පය තෝරාගනු ලැබේ.
3. Tossing a coin is an example. If the result is Head, display "Head"; otherwise, display "Tail."/ කාසියක් උඩ දැමීම ඊට උදාහරණයකි. ප්‍රතිඵලය හිස නම් "Head" ලෙස දර්ශනය කෙරේ; එසේ නොමැති නම් "Tail" ලෙස දර්ශනය කෙරේ.
4. Selection out of many options means selecting one option from more than two possible options./ බොහෝ විකල්ප අතරින් තේරීමක් කිරීම යනු විකල්ප දෙකකට වඩා වැඩි ගණනකින් එක් විකල්පයක් තෝරා ගැනීමයි.
5. Tossing a dice is an example. The value can be any number from 1 to 6, and a different action can be performed for each value./ දාළ කැටයක් වීසි කිරීම ඊට උදාහරණයකි. එහි අගය 1 සිට 6 දක්වා ඕනෑම අංකයක් විය හැකි අතර, එක් එක් අගය සඳහා වෙනස් ක්‍රියාවක් සිදු කළ හැකිය.
6. Repetition is an action that is performed again and again until a specified condition is no longer satisfied./ පුනරාවර්තනය යනු නිශ්චිත කොන්දේසියක් තවදුරටත් සපුරාලන්නේ නැති වන තුරු නැවත නැවතත් සිදු කරන ක්‍රියාවකි.
7. At the beginning, the condition is checked. If the condition is true, the repetitive work starts./ ආරම්භයේදී කොන්දේසිය පරීක්ෂා කරනු ලැබේ. කොන්දේසිය සත්‍ය නම්, පුනරාවර්තන ක්‍රියාවලිය ආරම්භ වේ.
8. When the condition becomes false, the repetition stops and the program proceeds to the next step or ends./ කොන්දේසිය අසත්‍ය වූ විට, පුනරාවර්තනය නතර වී වැඩසටහන ඊළඟ පියවරට යොමු වේ හෝ අවසන් වේ.
9. The pump is operated and water flows into the tank repeatedly. The condition "Is the tank full?" is checked: පොම්පය ක්‍රියාත්මක කර ටැංකියට ජලය පුරවන ක්‍රියාවලිය නැවත නැවත සිදු කෙරේ. "ටැංකිය පිරී තිබේද?" යන කොන්දේසිය පරීක්ෂා කෙරේ:
 - No → Continue filling the tank./ නැත → ටැංකිය පිරවීම දිගටම කරගෙන යන්න.
 - Yes → Stop the water pump./ ඔව් → ජල පොම්පය නවත්වන්න.
10. The student's name is called and the condition "Is the student present?" is checked: සිසුවාගේ නම කැඳවා "සිසුවා පැමිණ සිටීද?" යන කොන්දේසිය පරීක්ෂා කෙරේ:
 - Yes → Mark 1 for presence./ ඔව් → පැමිණීම සඳහා 1 ලෙස සලකුණු කරන්න.
 - No → Mark 0 for absence./ නැත → නොපැමිණීම සඳහා 0 ලෙස සලකුණු කරන්න.
 - Then the next student's name is called until all students are checked./ ඉන්පසු සියලු සිසුන් පරීක්ෂා කර අවසන් වන තුරු ඊළඟ සිසුවාගේ නම කැඳවනු ලැබේ.

11. Answer:

- Repetition with a definite number of times/ නිශ්චිත වාර ගණනක් පුනරාවර්තනය කිරීම
- Repeating until a given condition is satisfied/ යම් කොන්දේසියක් සපුරාලන තුරු පුනරාවර්තනය කිරීම
- Repeating an infinite number of times (Forever)/ අසීමිත වාර ගණනක් (සැමවිටම/Forever) පුනරාවර්තනය කිරීම

12. The Repeat block is used to execute a set of instructions a specified number of times./ 'Repeat' (පුනරාවර්තන) කොටස භාවිතා කරනුයේ උපදෙස් මාලාවක් නිශ්චිත වාර ගණනක් ක්‍රියාත්මක කිරීමටයි.

13. The Repeat Until block repeats instructions until a specified condition becomes true./ 'Repeat Until' (යම් කොන්දේසියක් සපුරාලන තුරු පුනරාවර්තනය) කොටස මගින් යම් කොන්දේසියක් සත්‍ය වන තුරු උපදෙස් පුනරාවර්තනය කරයි.

14. The Forever block repeats a set of instructions an infinite number of times, until the program is stopped./ 'Forever' (සැමවිටම) කොටස මගින් වැඩසටහන නතර කරන තුරු උපදෙස් මාලාවක් අසීමිත වාර ගණනක් පුනරාවර්තනය කරයි.

15. Start with the number of times as 1. Check whether it is ≤ 12 . If true, calculate:

Multiple = Number $\times$ Number of times

Then display the multiple and increase the number of times by 1. Repeat until the number of times becomes greater than 12./ 'වාර ගණන' 1 ලෙස ආරම්භ කරන්න. එය 12ට සමාන හෝ 12ට අඩු (12) දැයි පරීක්ෂා කරන්න. කොන්දේසිය සත්‍ය නම්, පහත පරිදි ගණනය කරන්න:

ගුණිතය = අංකය $\times$ වාර ගණන

ඉන්පසු එම ගුණිතය ප්‍රදර්ශනය කර, වාර ගණන 1කින් වැඩි කරන්න. වාර ගණන 12ට වඩා වැඩි වන තෙක් මෙම ක්‍රියාවලිය නැවත නැවත සිදු කරන්න.

16. Generating and displaying random numbers from 1 to 100 continuously is an example of an infinite repetition program./ 1 සිට 100 දක්වා අහඹු අංක අඛණ්ඩව ජනනය කිරීම සහ ප්‍රදර්ශනය කිරීම යනු අනන්ත පුනරාවර්තන වැඩසටහනකට උදාහරණයකි.

17. Nested repetition means placing one repetition structure inside another repetition structure./ කාවැද්දූ පුනරාවර්තනය යනු එක් පුනරාවර්තන ව්‍යුහයක් තවත් පුනරාවර්තන ව්‍යුහයක් තුළ තැබීමයි.

18. Creating a circular pattern using repeated movements and turns is an example. An outer repetition contains another Repeat block inside it./ නැවත නැවත සිදු කරන චලන සහ හැරීම් භාවිතයෙන් වෘත්තාකාර රටාවක් නිර්මාණය කිරීම ඊට උදාහරණයකි. මෙහිදී බාහිර පුනරාවර්තන ව්‍යුහය තුළ තවත් පුනරාවර්තන කොටසක් අඩංගු වේ.

19. Repetition with selection is a control structure where repetition is combined with conditional decisions. For example, in a Snakes and Ladders game, the program checks whether the player has reached a ladder or a snake and performs the appropriate action./ තෝරාගැනීම සහිත පුනරාවර්තනය යනු පුනරාවර්තනය සහ කොන්දේසි සහිත තීරණ ගැනීම් එකිනෙක සම්බන්ධ කෙරෙන පාලන ව්‍යුහයකි. උදාහරණයක් ලෙස, 'සර්පයන් සහ ඉණිමං' ක්‍රීඩාවකදී, ක්‍රීඩකයා ඉණිමංගතව හෝ සර්පයෙකු වෙත ළඟා වී ඇත්දැයි වැඩසටහන මගින් පරීක්ෂා කර ඊට අදාළ ක්‍රියාව සිදු කරයි.

20. The program: වැඩසටහන:

- Tosses a dice and gets its value./ දාදු කැටයක් දමා එහි අගය ලබා ගනී.
- Moves the counter forward according to the dice value./ දාදු කැටයේ අගයට අනුව ක්‍රීඩා ලකුණ ඉදිරියට ගෙන යයි.

#ICT කණ්ඩායම

- Checks whether the counter has reached the bottom of a ladder./ ක්‍රීඩා ලකුණ ඉණිමගක පහළ කෙළවරට ළඟා වී ඇත්දැයි පරීක්ෂා කරයි.
 - If yes, moves the counter to the top of the ladder./ එසේ නම්, ක්‍රීඩා ලකුණ ඉණිමගේ ඉහළ කෙළවරට ගෙන යයි.
 - Checks whether the counter has reached a snake's head./ ක්‍රීඩා ලකුණ සර්පයෙකුගේ හිස පිහිටි ස්ථානයට ළඟා වී ඇත්දැයි පරීක්ෂා කරයි.
 - If yes, moves the counter to the snake's tail./ එසේ නම්, ක්‍රීඩා ලකුණ සර්පයාගේ වලිගය පිහිටි ස්ථානයට ගෙන යයි.
 - Repeats until the counter reaches square 36./ ක්‍රීඩා ලකුණ 36 වන කොටුවට ළඟා වන තුරු මෙය නැවත නැවත සිදු කරයි.
 - Displays the end of the game./ ක්‍රීඩාව අවසන් බව පෙන්වයි.
21. An array is a data structure that can store many items using a single name./ අරේ එකක් යනු තනි නමක් භාවිතයෙන් අයිතම (දත්ත) රැසක් ගබඩා කළ හැකි දත්ත ව්‍යුහයකි.
22. Arrays reduce the need for many separate variables and make programs simpler and less complex, especially when storing a large number of values./ විශේෂයෙන් විශාල අගයන් ප්‍රමාණයක් ගබඩා කිරීමේදී, අරේ භාවිතය මගින් වෙන වෙනම විචල්‍යයන් විශාල සංඛ්‍යාවක් භාවිතා කිරීමේ අවශ්‍යතාව අඩු කරන අතර වැඩසටහන් වඩාත් සරල සහ සංකීර්ණ නොවන ඒවා බවට පත් කරයි.
23. An array can be created using a List in Scratch: Scratch හි 'List' (ලැයිස්තුව) භාවිතයෙන් අරේ එකක් නිර්මාණය කළ හැක:
- Select Make a List from the Data category./ 'Data' (දත්ත) කාණ්ඩය තුළින් 'Make a List' තෝරන්න.
 - Give the array a name./ අරේ එකට නමක් ලබා දෙන්න.
 - Select For this sprite only if required./ අවශ්‍ය නම් පමණක් 'For this sprite only' තෝරන්න.
 - Click OK./ 'OK' ක්ලික් කරන්න.
24. An array is called a List in Scratch programming./ Scratch ක්‍රමලේඛනයේදී අරේ එකක් 'List' (ලැයිස්තුව) ලෙස හැඳින්වේ.
25. Items can be added to a list using the add [item] to [list] block. For example, to enter five country names, the program can repeat five times, ask for the country name, and add the answer to the Country list./ 'add [item] to [list]' බ්ලොක් එක භාවිතයෙන් ලැයිස්තුවකට අයිතම එකතු කළ හැක. උදාහරණයක් ලෙස, රටවල් පහක නම් ඇතුළත් කිරීමට, වැඩසටහන මගින් පස් වතාවක් පුනරාවර්තනය සිදු කරමින් රටේ නම විමසා, එම පිළිතුර 'Country' ලැයිස්තුවට එකතු කළ හැක.
26. The program can use the length of [list] to repeat through all items and the item [number] of [list] block to display each item./ සියලුම අයිතම හරහා පුනරාවර්තනය වීමට 'length of [list]' ද, එක් එක් අයිතමය ප්‍රදර්ශනය කිරීමට 'item [number] of [list]' බ්ලොක් එක ද වැඩසටහනට භාවිතා කළ හැක.
27. Using individual variables requires a separate variable for each value, while an array stores multiple values under one name. Therefore, arrays can make programs shorter and easier to manage./ තනි තනි විචල්‍යයන් භාවිතා කිරීමේදී එක් එක් අගය සඳහා වෙනම විචල්‍යයක් අවශ්‍ය වන නමුත්, අරේ එකක් මගින් අගයන් කිහිපයක් තනි නමක් යටතේ ගබඩා කෙරේ. එබැවින්, අරේ භාවිතයෙන් වැඩසටහන් කෙටි සහ කළමනාකරණයට පහසු ඒවා බවට පත් කළ හැක.

28. Problem analysis means dividing a programming problem into smaller sections or steps so that the problem becomes easier to solve./ ගැටලුවක් විශ්ලේෂණය කිරීම යනු ගැටලුව විසඳීම පහසු කිරීම සඳහා ක්‍රමලේඛන ගැටලුවක් කුඩා කොටස්වලට හෝ පියවරවලට බෙදා වෙන් කිරීමයි.

29. The problem can be divided into four steps: මෙම ගැටලුව පියවර හතරකට බෙදා දැක්විය හැකිය:

- Input the ten numbers./ සංඛ්‍යා දහය ඇතුළත් කිරීම.
- Find the total of the ten numbers./ එම සංඛ්‍යා දහයේ එකතුව සොයා ගැනීම.
- Divide the total by 10 to find the average./ සාමාන්‍යය සොයා ගැනීම සඳහා එකතුව 10න් බෙදීම.
- Output the average./ සාමාන්‍යය ප්‍රතිදානය කිරීම.

30. Nested repetition is repetition placed inside another repetition. It is used when one repeated set of instructions needs to be performed within another repeated process./ කුඩුගත පුනරාවර්තනය යනු තවත් පුනරාවර්තනයක් තුළ පිහිටුවා ඇති පුනරාවර්තනයකි. තවත් පුනරාවර්තන ක්‍රියාවලියක් තුළ යම් පුනරාවර්තන උපදෙස් මාලාවක් ක්‍රියාත්මක කිරීමට අවශ්‍ය වූ විට මෙය භාවිතා කෙරේ.