13 ශ්‍රේණිය A/L ICT

පාඩම 9: ක්‍රමලේඛනය

- 9.1 ගැටලු විසඳීමේ ක්‍රියාවලිය
- 9.2 Top-down සැලසුම, පියවරෙන් පියවර පිරිපහදුව සහ ව්‍යුහ සටහන්
- 9.3 ඇල්ගොරිතම: ගැලීම් සටහන්, ව්‍යාජ කේත සහ අත් අනුරේඛන
- 9.4 ක්‍රමලේඛන භාෂා සහ ප්‍රතිමාන
- 9.5 වැඩසටහන් පරිවර්තනය: Compilers, Interpreters සහ Linkers
- 9.6 ඒකාබද්ධ සංවර්ධන පරිසර (IDE) සහ Debug කිරීම
- 9.7 Python මූලික: ව්‍යුහය, විචල්‍ය, දත්ත වර්ග, කාරක සහ ආදාන/ප්‍රතිදාන
- 9.8 පාලන ව්‍යුහ: අනුක්‍රමය, වරණය සහ පුනරාවර්තනය
- 9.9 ශ්‍රිත: ගොඩනැගූ සහ පරිශීලක නිර්වචිත
- 9.10 දත්ත ව්‍යුහ: තන්තු, ලැයිස්තු, Tuples සහ ශබ්දකෝෂ
- 9.11 ගොනු හැසිරවීම: විවෘත, කියවීම, ලිවීම, එකතු කිරීම සහ වැසීම
- 9.12 Python සහ දත්ත සමුදා: සම්බන්ධ, එකතු, වෙනස්, මැකීම, ලබා ගැනීම
- 9.13 සෙවීම සහ වර්ග කිරීම: අනුක්‍රමික සෙවීම සහ Bubble Sort

© 2026 schoolict.net (Ruwan Mangala Suraweera)
සියලු හිමිකම් ඇවිරිණි.

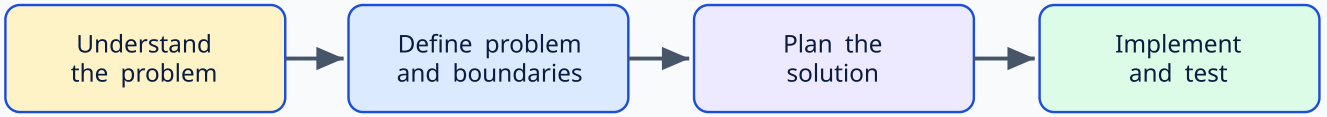
13 ශ්‍රේණිය A/L ICT · පාඩම 9

9.1 ගැටලු විසඳීමේ ක්‍රියාවලිය

විෂය නිර්දේශය 9.1. කේත ලිවීමට පෙර ක්‍රමලේඛකයා ගැටලුව කඩදාසිය මත විසඳයි. පියවර හතර: **ගැටලුව තේරුම් ගැනීම**, **ගැටලුව සහ එහි සීමා නිර්වචනය**, **විසඳුම සැලසුම් කිරීම** සහ එය **ක්‍රියාත්මක කිරීම**.

1. පියවර හතර

Problem-solving process



Input → process → output: list what is given, what is needed and the limits first.

ගැටලු විසඳීමේ පියවර

තේරුම් ගැනීම
ලබා දී ඇත්තේ (ආදානය) සහ අවශ්‍ය වන්නේ (ප්‍රතිදානය) කුමක්ද?

නිර්වචනය සහ සීමා
නිශ්චිත කාර්යය සහ සීමා; උදා: දිග මීටර්වලින් ධන සංඛ්‍යා.

සැලසුම
පියවර (ඇල්ගොරිතමය) ගැලීම් සටහනක් හෝ pseudocode ලෙස.

ක්‍රියාත්මක කිරීම
වැඩසටහන ලියා පරීක්ෂණ දත්ත සමඟ ධාවනය කර දෝෂ නිවැරදි කර, එය ගැටලුව විසඳන්නේදැයි ඇගයීම.

කොටස	සෘජුකෝණාස්‍ර ගැටලුව
ආදානය	length, width
සැකසුම	area = length × width; perimeter = 2 × (length + width)
ප්‍රතිදානය	area, perimeter
සීමාව	length සහ width > 0

2. සැලසුම Python මගින් ක්‍රියාත්මක කිරීම

🐍 සෘජුකෝණාස්‍රයේ වර්ගඵලය සහ පරිමිතිය

```
# 1 input  2 process  3 output
length = float(input("Length: "))
width = float(input("Width: "))
area = length * width
perimeter = 2 * (length + width)
print("Area =", area)
print("Perimeter =", perimeter)
```

▶ **ප්‍රතිදානය**

```
Length: 5
Width: 3
Area = 15.0
Perimeter = 16.0
```

🧱 **විසඳූ උදාහරණය: length 5, width 3 සමඟ වැඩසටහන පරීක්ෂා කරන්න.** අතින් ගණනය: වර්ගඵලය = **15**, පරිමිතිය = **16**. වැඩසටහන Area = 15.0 සහ Perimeter = 16.0 මුද්‍රණය කළ නිසා සැලසුම නිවැරදිය.

💡 වැඩසටහන ධාවනයට පෙර බලාපොරොත්තු පිළිතුර අතින් ගණනය කරන්න.

🚀 **විෂය නිර්දේශයෙන් ඔබ්බට (2026 යාවත්කාලීනය)** *විභාගයට අවශ්‍ය නැත – දැනුම සඳහා කියවන්න.* 2026 දී AI සහායකයින් ගැටලු ප්‍රකාශයෙන් කේත ලියයි; නමුත් පැහැදිලි නිර්වචනය සහ පරීක්ෂණ අවස්ථා ක්‍රමලේඛකයාගේ කාර්යයයි.

ප්‍රධාන පද

ගැටලු විසඳීම – ඉලක්කයකට ළඟා වීමේ ක්‍රමයක් සෙවීම.

ගැටලුව තේරුම් ගැනීම – ලබා දී ඇති දේ සහ අවශ්‍ය දේ සෙවීම.

ගැටලු නිර්වචනය – විසඳිය යුතු කාර්යයේ නිශ්චිත ප්‍රකාශය.

සීමාව – ගැටලුවක විෂය පථයේ හෝ අවසර අගයන්හි සීමාව.

සැලසුම් කිරීම – කේතනයට පෙර විසඳුමේ පියවර සැලසුම් කිරීම.

ක්‍රියාත්මක කිරීම – සැලසුම් කළ විසඳුම කේතනය, ධාවනය සහ පරීක්ෂාව.

ආදානය – විසඳුමට ලබා දෙන දත්ත.

සැකසුම – ආදානය ප්‍රතිදානය බවට හරවන පියවර.

ප්‍රතිදානය – විසඳුම නිපදවන ප්‍රතිඵලය.

පරීක්ෂණ දත්ත – වැඩසටහනක් පරීක්ෂා කිරීමට ගන්නා ආදාන.

බලාපොරොත්තු ප්‍රතිඵලය – ධාවනයට පෙර අතින් ගණනය කළ පිළිතුර.

ඇල්ගොරිතමය – ගැටලුවක් විසඳන සීමිත පිළිවෙළ පියවර.

ආදර්ශ ප්‍රශ්න සහ පිළිතුරු

1. පියවර හතර?

> **පිළිතුර:** තේරුම් ගැනීම, නිර්වචනය, සැලසුම, ක්‍රියාත්මක කිරීම.

2. සීමාව?

> **පිළිතුර:** “length > 0” වැනි සීමාවක්.

3. ආදාන?

> **පිළිතුර:** length සහ width.

4. 5, 3 ප්‍රතිදානය?

> **පිළිතුර:** 15.0, 16.0.

5. කේතනයට පෙර සැලසුම ඇයි?

> **පිළිතුර:** අඩු දෝෂ, පැහැදිලි පියවර.

6. ක්‍රියාත්මක කිරීමේදී?

> **පිළිතුර:** කේතනය, ධාවනය, debug.

බහුවරණ ප්‍රශ්න

1. පළමු පියවර

A. printing output
B. writing code
C. understanding the problem
D. choosing a language
E. testing

2. ධන විය යුතුය

A. process
B. test result
C. output
D. comment
E. boundary of the problem

3. ගැලීම් සටහන

A. planning
B. understanding
C. maintenance
D. output
E. implementation

4. වර්ගඵලය

A. 16.0
B. 8.0
C. 2.0
D. 15.0
E. 30.0

5. පරිමිතිය

A. 30.0
B. 8.0
C. 16.0
D. 15.0
E. 10.0

6. නියැදි අගය

A. boundary setting
B. implementation
C. planning only
D. understanding
E. defining

7. length, width

A. inputs
B. outputs
C. comments
D. constants only
E. boundaries

8. අතින්

A. boundary
B. actual output
C. expected result
D. syntax error
E. variable

9. “ඉලක්කයකට ළඟා වීමේ ක්‍රමයක් සෙවීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ගැටලු නිර්වචනය
B. බලාපොරොත්තු ප්‍රතිඵලය
C. සැකසුම
D. ක්‍රියාත්මක කිරීම
E. ගැටලු විසඳීම

10. “ලබා දී ඇති දේ සහ අවශ්‍ය දේ සෙවීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. සීමාව
B. ගැටලුව තේරුම් ගැනීම
C. ආදානය
D. ඇල්ගොරිතමය
E. ප්‍රතිදානය

11. “විසඳිය යුතු කාර්යයේ නිශ්චිත ප්‍රකාශය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. පරීක්ෂණ දත්ත
B. සැකසුම
C. ගැටලු විසඳීම
D. සැලසුම් කිරීම
E. ගැටලු නිර්වචනය

12. “ගැටලුවක විෂය පථයේ හෝ අවසර අගයන්හි සීමාව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ප්‍රතිදානය
B. ගැටලුව තේරුම් ගැනීම
C. බලාපොරොත්තු ප්‍රතිඵලය
D. සීමාව
E. ක්‍රියාත්මක කිරීම

13. “කේතනයට පෙර විසඳුමේ පියවර සැලසුම් කිරීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. පරීක්ෂණ දත්ත
B. ආදානය
C. ඇල්ගොරිතමය
D. සැලසුම් කිරීම
E. ගැටලු නිර්වචනය

14. “සැලසුම් කළ විසඳුම කේතනය, ධාවනය සහ පරීක්ෂාව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ක්‍රියාත්මක කිරීම
B. සැකසුම
C. බලාපොරොත්තු ප්‍රතිඵලය
D. ගැටලු විසඳීම
E. සීමාව

15. "විසඳුමට ලබා දෙන දත්ත" යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ආදානය
B. ඇල්ගොරිතමය
C. සැලසුම් කිරීම
D. ගැටලුව තේරුම් ගැනීම
E. ප්‍රතිදානය

16. "ආදානය ප්‍රතිදානය බවට හරවන පියවර" යන්නෙන් අදහස් වන පදය කුමක්ද?

A. පරීක්ෂණ දත්ත
B. ගැටලු නිර්වචනය
C. ක්‍රියාත්මක කිරීම
D. ගැටලු විසඳීම
E. සැකසුම

17. "විසඳුම නිපදවන ප්‍රතිඵලය" යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ආදානය
B. ප්‍රතිදානය
C. බලාපොරොත්තු ප්‍රතිඵලය
D. ගැටලුව තේරුම් ගැනීම
E. සීමාව

18. "වැඩසටහනක් පරීක්ෂා කිරීමට ගන්නා ආදාන" යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ගැටලු නිර්වචනය
B. සැලසුම් කිරීම
C. පරීක්ෂණ දත්ත
D. ඇල්ගොරිතමය
E. සැකසුම

19. "ධාවනයට පෙර අතින් ගණනය කළ පිළිතුර" යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ගැටලු විසඳීම
B. බලාපොරොත්තු ප්‍රතිඵලය
C. ප්‍රතිදානය
D. ක්‍රියාත්මක කිරීම
E. සීමාව

20. "ගැටලුවක් විසඳන සීමිත පිළිවෙළ පියවර" යන්නෙන් අදහස් වන පදය කුමක්ද?

A. පරීක්ෂණ දත්ත
B. ආදානය
C. ගැටලුව තේරුම් ගැනීම
D. ඇල්ගොරිතමය
E. සැලසුම් කිරීම

ව්‍යුහගත ප්‍රශ්න

1. පියවර හතර. *(ලකුණු 4)*

> තේරුම්; නිර්වචනය; සැලසුම; ක්‍රියාත්මක.

2. ආදාන, සැකසුම, ප්‍රතිදාන. *(ලකුණු 3)*

> ආදාන; සැකසුම; ප්‍රතිදාන.

3. සීමා දෙකක්. *(ලකුණු 2)*

> 0–100; පූර්ණ සංඛ්‍යා.

4. බලාපොරොත්තු ප්‍රතිඵල ඇයි? *(ලකුණු 2)*

> සැසඳීම; තාර්කික දෝෂ.

5. Python වැඩසටහනක්. *(ලකුණු 3)*

> input; සූත්‍ර; print.

සාරාංශය

- මුලින් ගැටලුව තේරුම් ගන්න.
- කාර්යය සහ සීමා නිර්වචනය කරන්න.
- කේතනයට පෙර පියවර සැලසුම් කරන්න.
- දත්ත සමඟ ක්‍රියාත්මක කර පරීක්ෂා කරන්න.
- ප්‍රතිදානය බලාපොරොත්තු ප්‍රතිඵලය සමඟ සසඳන්න.

MCQ පිළිතුරු – 9.1

1. **C** – මුලින් තේරුම් ගැනීම.
2. **E** – සීමාව.
3. **A** – සැලසුම.
4. **D** – 5 × 3.
5. **C** – 2 × 8.
6. **B** – ක්‍රියාත්මක කිරීම.
7. **A** – ආදාන.
8. **C** – බලාපොරොත්තු.
9. **E** – ගැටලු විසඳීම – ඉලක්කයකට ළඟා වීමේ ක්‍රමයක් සෙවීම.
10. **B** – ගැටලුව තේරුම් ගැනීම – ලබා දී ඇති දේ සහ අවශ්‍ය දේ සෙවීම.
11. **E** – ගැටලු නිර්වචනය – විසඳිය යුතු කාර්යයේ නිශ්චිත ප්‍රකාශය.
12. **D** – සීමාව – ගැටලුවක විෂය පථයේ හෝ අවසර අගයන්හි සීමාව.
13. **D** – සැලසුම් කිරීම – කේතනයට පෙර විසඳුමේ පියවර සැලසුම් කිරීම.
14. **A** – ක්‍රියාත්මක කිරීම – සැලසුම් කළ විසඳුම කේතනය, ධාවනය සහ පරීක්ෂාව.
15. **A** – ආදානය – විසඳුමට ලබා දෙන දත්ත.

16. **E** – සැකසුම – ආදානය ප්‍රතිදානය බවට හරවන පියවර.
17. **B** – ප්‍රතිදානය – විසඳුම නිපදවන ප්‍රතිඵලය.
18. **C** – පරීක්ෂණ දත්ත – වැඩසටහනක් පරීක්ෂා කිරීමට ගන්නා ආදාන.
19. **B** – බලාපොරොත්තු ප්‍රතිඵලය – ධාවනයට පෙර අතින් ගණනය කළ පිළිතුර.
20. **D** – ඇල්ගොරිතමය – ගැටලුවක් විසඳන සීමිත පිළිවෙළ පියවර.

© 2026 schoolict.net (Ruwan Mangala Suraweera). සියලු හිමිකම් ඇවිරිණි.

13 ශ්‍රේණිය A/L ICT · පාඩම 9

9.2 Top-down සැලසුම, පියවරෙන් පියවර පිරිපහදුව සහ ව්‍යුහ සටහන්

විෂය නිර්දේශය 9.2. විශාල ගැටලු කුඩා කොටස්වලට බෙදූ විට පහසුය. **මොඩියුලකරණය** වැඩසටහනක් මොඩියුලවලට බෙදයි; **top-down සැලසුම** සම්පූර්ණ කාර්යයෙන් ආරම්භ කර **පියවරෙන් පියවර පිරිපහදුව** එක් එක් පියවර කේතනය කළ හැකි තෙක් කුඩා කරයි. **ව්‍යුහ සටහන** මොඩියුල සහ ඒවා එකිනෙක කැඳවන ආකාරය පෙන්වයි.

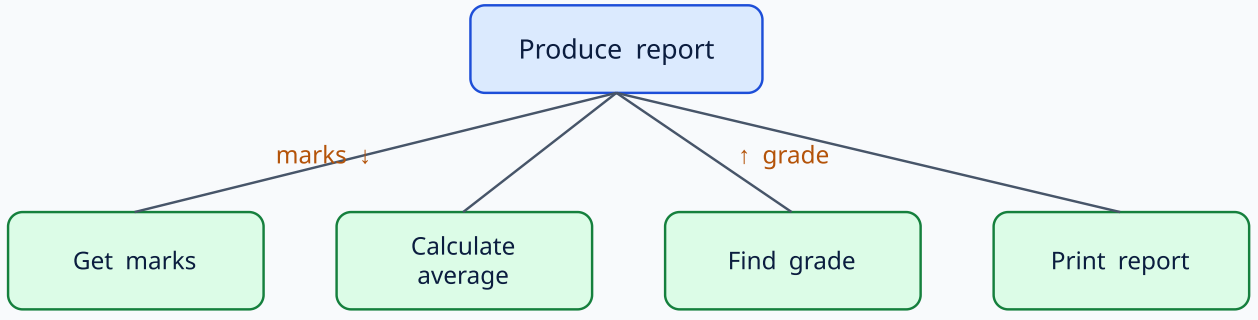
1. පියවරෙන් පියවර පිරිපහදුව

කාර්යය පියවරෙන් පියවර පිරිපහදු කිරීම

```
1 Produce a class report
  1.1 Get the marks
  1.2 Calculate the average
      1.2.1 Add all marks
      1.2.2 Divide by the number of marks
  1.3 Find the grade
      1.3.1 IF average >= 75 THEN A
      1.3.2 ELSE IF average >= 50 THEN C ELSE F
  1.4 Print the report
```

2. ව්‍යුහ සටහන

Structure chart (top-down design)



Each box is a module; lines show which module calls which; arrows on lines show data passed.

පන්ති වාර්තාව සඳහා ව්‍යුහ සටහන

- සෑම **සෘජුකෝණාස්‍රයක්ම** මොඩියුලයකි (Python හි ශ්‍රිතයක්).
- **රේඛා** කැඳවන මොඩියුලය කැඳවනු ලබන ඒවාට යා කරයි.
- කුඩා ඊතල පහළට (marks) සහ ඉහළට (grade) යවන **දත්ත** පෙන්වයි.

3. සෑම මොඩියුලයක්ම ශ්‍රිතයක් වේ

🐍 Top-down: කුඩා ශ්‍රිතවලින් සාදන වාර්තාව

```
def get_marks():
    return [78, 45, 88, 62, 39]

def average(marks):
    return sum(marks) / len(marks)

def grade(avg):
    if avg >= 75:
        return "A"
    elif avg >= 50:
        return "C"
    return "F"

def report():
    marks = get_marks()
    avg = average(marks)
    print("Average:", avg)
    print("Grade:", grade(avg))

report()
```

▶ **ප්‍රතිදානය**

```
Average: 62.4
Grade: C
```

🧱 **විසඳූ උදාහරණය: මොඩියුල වැඩසටහන මුද්‍රණය කරන්නේ කුමක්ද?**
average() = **62.4**; grade() 62.4 ≥ 50 නමුත් < 75 නිසා **C**; report() දෙකම මුද්‍රණය කරයි.

💡 මොඩියුලවල වාසි: පහසු පරීක්ෂාව, නැවත භාවිතය, කණ්ඩායම් වැඩ සහ පහසු නඩත්තුව.

🚀 විෂය නිර්දේශයෙන් ඔබ්බට (2026 යාවත්කාලීනය) *විභාගයට අවශ්‍ය නැත – දැනුම සඳහා කියවන්න.* 2026 විශාල යෙදුම් microservices ලෙස බෙදා ඇත – එකම top-down අදහස සේවාදායක මට්ටමින්.

ප්‍රධාන පද

මොඩියුලකරණය – වැඩසටහනක් වෙන් මොඩියුලවලට බෙදීම.

මොඩියුලය – එක් කාර්යයක් කරන ස්වාධීන කොටස.

Top-down සැලසුම – සම්පූර්ණ ගැටලුවෙන් ආරම්භ කර කොටස්වලට බෙදීම.

පියවරෙන් පියවර පිරිපහදුව – සෑම පියවරක්ම කුඩා විස්තරාත්මක පියවරවලට බෙදීම.

ව්‍යුහ සටහන – මොඩියුල සහ කැඳවීමේ ධුරාවලියේ රූපය.

ප්‍රධාන මොඩියුලය – උප මොඩියුල පාලනය කරන ඉහළම මොඩියුලය.

උප මොඩියුලය – ඉහළ මොඩියුලයක් කැඳවන මොඩියුලය.

දත්ත යුගලය – යවන දත්ත පෙන්වන ව්‍යුහ සටහනේ ඊතලය.

නැවත භාවිතය – එකම මොඩියුලය ස්ථාන කිහිපයක භාවිතය.

ශ්‍රිතය – කැඳවිය හැකි නම් කළ කේත කොටස.

Bottom-up සැලසුම – කුඩා කොටස් මුලින් සාදා එකතු කිරීම.

වියෝජනය – සංකීර්ණ ගැටලුවක් සරල ඒවාට බිඳීම.

ආදර්ශ ප්‍රශ්න සහ පිළිතුරු

1. පියවරෙන් පියවර පිරිපහදුව?

පිළිතුර: කේතනය කළ හැකි තෙක් පියවර කුඩා කිරීම.

2. ව්‍යුහ සටහන?

පිළිතුර: මොඩියුල, කැඳවීම්, දත්ත.

3. වාසි දෙකක්?

පිළිතුර: පහසු පරීක්ෂාව සහ නැවත භාවිතය.

4. සාමාන්‍යය?

පිළිතුර: 62.4.

5. ශ්‍රේණිය?

පිළිතුර: C.

6. Top-down සහ bottom-up?

පිළිතුර: Top-down බෙදයි; bottom-up එකතු කරයි.

බහුවරණ ප්‍රශ්න

1. සම්පූර්ණයෙන් ආරම්භ

A. top-down design
B. compiling
C. hand tracing
D. bottom-up design
E. debugging

2. කුඩා පියවර

A. sorting
B. stepwise refinement
C. normalisation
D. interpretation
E. linking

3. මොඩියුල රූපය

A. structure chart
B. trace table
C. flowchart symbol
D. Gantt chart
E. ER diagram

4. Python

A. tuple
B. comment
C. function
D. constant
E. file mode

5. සාමාන්‍යය

A. 60.4
B. 62.4
C. 65.0
D. 62.0
E. 78.0

6. ශ්‍රේණිය

A. S
B. F
C. A
D. C
E. B

7. ඊතල

A. data passed between modules
B. loop counters
C. file names
D. comments
E. syntax errors

8. වාසියක් නොවේ

A. easier maintenance
B. reuse of code
C. teamwork
D. easier testing
E. it always makes programs run faster

9. “වැඩසටහනක් වෙන් මොඩියුලවලට බෙදීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. මොඩියුලකරණය
B. දත්ත යුගලය
C. ප්‍රධාන මොඩියුලය
D. Top-down සැලසුම
E. Bottom-up සැලසුම

10. “එක් කාර්යයක් කරන ස්වාධීන කොටස” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. වියෝජනය
B. මොඩියුලය
C. පියවරෙන් පියවර පිරිපහදුව
D. නැවත භාවිතය
E. උප මොඩියුලය

11. “සම්පූර්ණ ගැටලුවෙන් ආරම්භ කර කොටස්වලට බෙදීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. මොඩියුලකරණය
B. ශ්‍රිතය
C. ව්‍යුහ සටහන
D. Top-down සැලසුම
E. දත්ත යුගලය

12. “සෑම පියවරක්ම කුඩා විස්තරාත්මක පියවරවලට බෙදීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ප්‍රධාන මොඩියුලය
B. Bottom-up සැලසුම
C. මොඩියුලය
D. නැවත භාවිතය
E. පියවරෙන් පියවර පිරිපහදුව

13. “මොඩියුල සහ කැඳවීමේ ධුරාවලියේ රූපය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. උප මොඩියුලය
B. Top-down සැලසුම
C. වියෝජනය
D. ශ්‍රිතය
E. ව්‍යුහ සටහන

14. “උප මොඩියුල පාලනය කරන ඉහළම මොඩියුලය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. පියවරෙන් පියවර පිරිපහදුව
B. ප්‍රධාන මොඩියුලය
C. දත්ත යුගලය
D. Bottom-up සැලසුම
E. මොඩියුලකරණය

15. “ඉහළ මොඩියුලයක් කැඳවන මොඩියුලය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. වියෝජනය
B. ව්‍යුහ සටහන
C. නැවත භාවිතය
D. උප මොඩියුලය
E. මොඩියුලය

16. “යවන දත්ත පෙන්වන ව්‍යුහ සටහනේ ඊතලය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. මොඩියුලකරණය
B. Top-down සැලසුම
C. දත්ත යුගලය
D. ප්‍රධාන මොඩියුලය
E. ශ්‍රිතය

17. “එකම මොඩියුලය ස්ථාන කිහිපයක භාවිතය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. උප මොඩියුලය
B. Bottom-up සැලසුම
C. පියවරෙන් පියවර පිරිපහදුව
D. නැවත භාවිතය
E. මොඩියුලය

18. “කැඳවිය හැකි නම් කළ කේත කොටස” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. වියෝජනය
B. Top-down සැලසුම
C. ශ්‍රිතය
D. දත්ත යුගලය
E. ව්‍යුහ සටහන

19. “කුඩා කොටස් මුලින් සාදා එකතු කිරීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. මොඩියුලකරණය
B. පියවරෙන් පියවර පිරිපහදුව
C. Bottom-up සැලසුම
D. නැවත භාවිතය
E. ප්‍රධාන මොඩියුලය

20. “සංකීර්ණ ගැටලුවක් සරල ඒවාට බිඳීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ශ්‍රිතය
B. ව්‍යුහ සටහන
C. උප මොඩියුලය
D. මොඩියුලය
E. වියෝජනය

ව්‍යුහගත ප්‍රශ්න

1. Top-down සහ පිරිපහදුව. *(ලකුණු 3)*

> සම්පූර්ණ → කොටස්; පිරිපහදුව.

2. "සාමාන්‍යය" පිරිපහදුව. *(ලකුණු 3)*

> එකතු; ගණන් කිරීම; බෙදීම.

3. ව්‍යුහ සටහන. *(ලකුණු 3)*

> ප්‍රධාන; උප මොඩියුල 4; දත්ත.

4. වාසි තුනක්. *(ලකුණු 3)*

> පරීක්ෂාව; නැවත භාවිතය; නඩත්තුව.

5. grade() ශ්‍රිතය. *(ලකුණු 4)*

> def; if; elif; return.

සාරාංශය

- මොඩියුලකරණය වැඩසටහන් මොඩියුලවලට බෙදයි.
- Top-down: මුලින් සම්පූර්ණ ගැටලුව.
- පිරිපහදුව පියවරෙන් පියවර විස්තර එක් කරයි.
- ව්‍යුහ සටහන් මොඩියුල සහ කැඳවීම් පෙන්වයි.
- සෑම මොඩියුලයක්ම ශ්‍රිතයක් වේ.

MCQ පිළිතුරු – 9.2

1. **A** – Top-down.
2. **B** – පිරිපහදුව.
3. **A** – ව්‍යුහ සටහන.
4. **C** – ශ්‍රිතය.
5. **B** – 312 ÷ 5.
6. **D** – 50 ≤ 62.4 < 75.
7. **A** – දත්ත යුගල.
8. **E** – වේගය සහතික නැත.
9. **A** – මොඩියුලකරණය – වැඩසටහනක් වෙන් මොඩියුලවලට බෙදීම.
10. **B** – මොඩියුලය – එක් කාර්යයක් කරන ස්වාධීන කොටස.
11. **D** – Top-down සැලසුම – සම්පූර්ණ ගැටලුවෙන් ආරම්භ කර කොටස්වලට බෙදීම.
12. **E** – පියවරෙන් පියවර පිරිපහදුව – සෑම පියවරක්ම කුඩා විස්තරාත්මක පියවරවලට බෙදීම.
13. **E** – ව්‍යුහ සටහන – මොඩියුල සහ කැඳවීමේ ධුරාවලියේ රූපය.
14. **B** – ප්‍රධාන මොඩියුලය – උප මොඩියුල පාලනය කරන ඉහළම මොඩියුලය.
15. **D** – උප මොඩියුලය – ඉහළ මොඩියුලයක් කැඳවන මොඩියුලය.

16. **C** – දත්ත යුගලය – යවන දත්ත පෙන්වන ව්‍යුහ සටහනේ ඊතලය.
17. **D** – නැවත භාවිතය – එකම මොඩියුලය ස්ථාන කිහිපයක භාවිතය.
18. **C** – ශ්‍රිතය – කැඳවිය හැකි නම් කළ කේත කොටස.
19. **C** – Bottom-up සැලසුම – කුඩා කොටස් මුලින් සාදා එකතු කිරීම.
20. **E** – වියෝජනය – සංකීර්ණ ගැටලුවක් සරල ඒවාට බිඳීම.

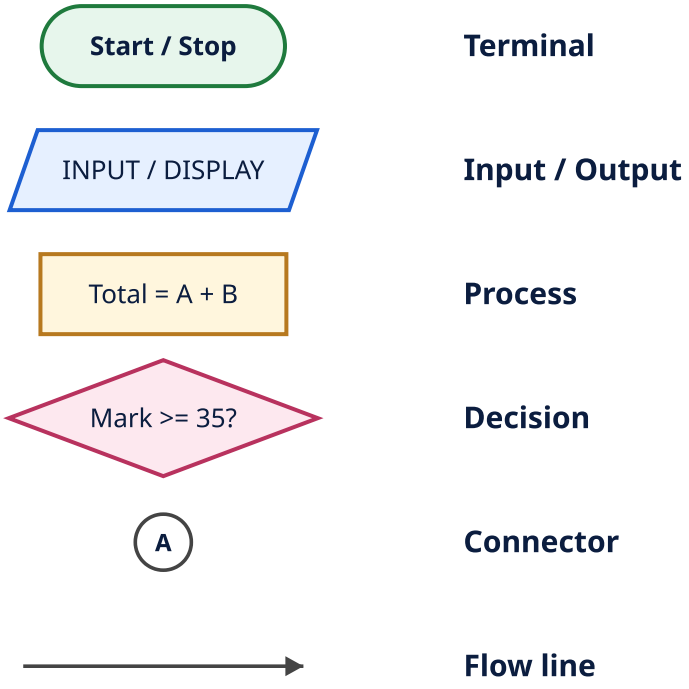
© 2026 schoolict.net (Ruwan Mangala Suraweera). සියලු හිමිකම් ඇවිරිණි.

13 ශ්‍රේණිය A/L ICT · පාඩම 9

9.3 ඇල්ගොරිතම: ගැලීම් සටහන්, ව්‍යාජ කේත සහ අත් අනුරේඛන

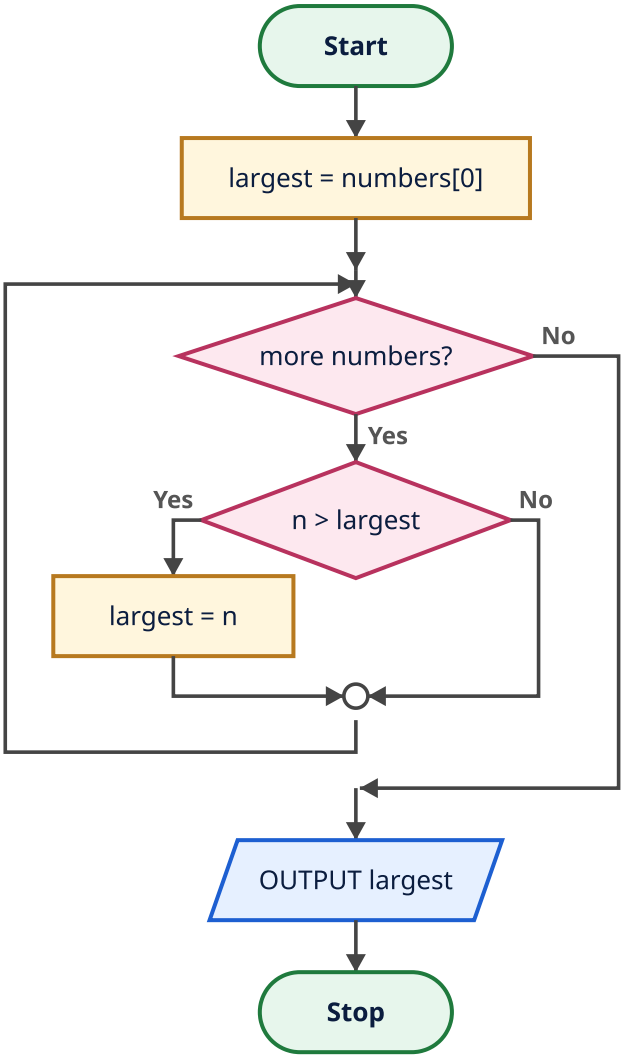
විෂය නිර්දේශය 9.3. ඇල්ගොරිතමයක් යනු ගැටලුවක් විසඳන සීමිත, පිළිවෙළ, පැහැදිලි පියවර කට්ටලයකි. අපි ඒවා **ගැලීම් සටහන්** (සම්මත සංකේත) හෝ **ව්‍යාජ කේත** ලෙස පෙන්වා කේතනයට පෙර **අත් අනුරේඛනයක්** (trace table) මගින් පරීක්ෂා කරමු.

1. ගැලීම් සටහන් සංකේත



සම්මත ගැලීම් සටහන් සංකේත

2. විශාලතම සංඛ්‍යාව සෙවීම



ගැලීම් සටහන: විශාලතම සංඛ්‍යාව

ව්‍යාජ කේතය (Pseudocode)

```
BEGIN
  largest = first number
  FOR each number n DO
    IF n > largest THEN
      largest = n
    ENDIF
  ENDFOR
  DISPLAY largest
END
```

3. අත් අනුරේඛනය

n	n > largest?	largest
12	False	12
45	True	45

7	False	45
30	False	45

🧱 **විසඳූ උදාහරණය: [12, 45, 7, 30] සඳහා ඇල්ගොරිතමය අනුරේඛනය කරන්න.**
largest 12 සිට ආරම්භ වේ. 45 පමණක් විශාල නිසා largest 45 වේ. ප්‍රතිදානය **45** – පහත Python වැඩසටහනද එයම මුද්‍රණය කරයි.

🐍 විශාලතම සංඛ්‍යාව සෙවීම

```
numbers = [12, 45, 7, 30]
largest = numbers[0]
for n in numbers:
    if n > largest:
        largest = n
print("Largest:", largest)
```

▶ ප්‍රතිදානය

```
Largest: 45
```

💡 trace table එකක විචල්‍යයකට තීරුවක් සහ අගය වෙනස් වන සෑම විටම පේළියක් ඇත.

🚀 **විෂය නිර්දේශයෙන් ඔබ්බට (2026 යාවත්කාලීනය)** *විභාගයට අවශ්‍ය නැත – දැනුම සඳහා කියවන්න.* 2026 visual debuggers ස්වයංක්‍රීයව trace tables අඳියි; නමුත් විභාගයේ අතින් අනුරේඛනය කළ යුතුය.

ප්‍රධාන පද

ඇල්ගොරිතමය – ගැටලුවක් විසඳන සීමිත පිළිවෙළ පැහැදිලි පියවර.

ගැලීම් සටහන – සම්මත සංකේත භාවිත කරන ඇල්ගොරිතම රූපය.

ව්‍යාජ කේතය – ඇල්ගොරිතමයක ව්‍යුහගත ඉංග්‍රීසි විස්තරය.

අත් අනුරේඛනය – අගයන් පරීක්ෂාවට ඇල්ගොරිතමය අතින් ක්‍රියාත්මක කිරීම.

Trace table – අනුරේඛනයේදී විචල්‍ය අගයන් සටහන් කරන වගුව.

අන්ත සංකේතය – ආරම්භය හෝ නැවතුම පෙන්වන රවුම් කොටුව.

සැකසුම් සංකේතය – ගණනය හෝ පැවරුම සඳහා සෘජුකෝණාස්‍රය.

තීරණ සංකේතය – ඔව්/නැත කොන්දේසියක් සහිත දියමන්තිය.

ආදාන/ප්‍රතිදාන සංකේතය – දත්ත කියවීම/පෙන්වීම සඳහා සමාන්තරාස්‍රය.

ගැලීම් රේඛාව – පියවර පිළිවෙළ පෙන්වන ඊතලය.

සම්බන්ධකය – ගැලීම් සටහනේ කොටස් යා කරන කුඩා කවය.

වියළි ධාවනය – පරිගණකයක් නැතිව අනුරේඛනයට තවත් නමක්.

ආදර්ශ ප්‍රශ්න සහ පිළිතුරු

1. ඇල්ගොරිතමය?

> **පිළිතුර:** සීමිත පියවෙළ පැහැදිලි පියවර.

2. තීරණ සංකේතය?

> **පිළිතුර:** දියමන්තිය.

3. ආදාන/ප්‍රතිදාන?

> **පිළිතුර:** සමාන්තරාස්‍රය.

4. ප්‍රතිදානය?

> **පිළිතුර:** 45.

5. largest වෙනස් වන වාර?

> **පිළිතුර:** එක් වරක්.

6. අත් අනුරේඛනය ඇයි?

> **පිළිතුර:** කේතනයට පෙර තාර්කික දෝෂ සෙවීමට.

බහුවරණ ප්‍රශ්න

1. දියමන්තිය

A. connector
B. decision
C. terminal
D. input
E. process

2. සමාන්තරාස්‍රය

A. a connector
B. start or stop
C. input or output
D. a calculation only
E. a decision

3. රවුම්

A. a loop counter
B. input
C. a decision
D. start or stop
E. a file

4. ව්‍යුහගත ඉංග්‍රීසි

A. a trace table
B. pseudocode
C. bytecode
D. a flowchart
E. machine code

5. විශාලතම

A. 94
B. 7
C. 30
D. 12
E. 45

6. True

A. 12 only
B. all values
C. no value
D. 7 and 30
E. 45 only

7. වගුව

A. trace table
B. hash table
C. symbol table
D. data dictionary
E. truth table

8. ඇල්ගොරිතමය

A. random
B. unordered
C. finite and unambiguous
D. written only in Python
E. infinite

9. “ගැටලුවක් විසඳන සීමිත පිළිවෙළ පැහැදිලි පියවර” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ඇල්ගොරිතමය
B. තීරණ සංකේතය
C. අන්ත සංකේතය
D. සම්බන්ධකය
E. ව්‍යාජ කේතය

10. “සම්මත සංකේත භාවිත කරන ඇල්ගොරිතම රූපය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. වියළි ධාවනය
B. ගැලීම් සටහන
C. සැකසුම් සංකේතය
D. අත් අනුරේඛනය
E. ආදාන/ප්‍රතිදාන සංකේතය

11. “ඇල්ගොරිතමයක ව්‍යුහගත ඉංග්‍රීසි විස්තරය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ව්‍යාජ කේතය
B. Trace table
C. ඇල්ගොරිතමය
D. තීරණ සංකේතය
E. ගැලීම් රේඛාව

12. “අගයන් පරීක්ෂාවට ඇල්ගොරිතමය අතින් ක්‍රියාත්මක කිරීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. අන්ත සංකේතය
B. සම්බන්ධකය
C. ආදාන/ප්‍රතිදාන සංකේතය
D. ගැලීම් සටහන
E. අත් අනුරේඛනය

13. “අනුරේඛනයේදී විචල්‍ය අගයන් සටහන් කරන වගුව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ව්‍යාජ කේතය
B. සැකසුම් සංකේතය
C. ගැලීම් රේඛාව
D. Trace table
E. වියළි ධාවනය

14. “ආරම්භය හෝ නැවතුම පෙන්වන රවුම් කොටුව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. සම්බන්ධකය
B. තීරණ සංකේතය
C. අත් අනුරේඛනය
D. අන්ත සංකේතය
E. ඇල්ගොරිතමය

15. “ගණනය හෝ පැවරුම සඳහා සෘජුකෝණාස්‍රය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ගැලීම් සටහන
B. Trace table
C. සැකසුම් සංකේතය
D. වියළි ධාවනය
E. ආදාන/ප්‍රතිදාන සංකේතය

16. “ඔව්/නැත කොන්දේසියක් සහිත දියමන්තිය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. තීරණ සංකේතය
B. ගැලීම් රේඛාව
C. අන්ත සංකේතය
D. ව්‍යාජ කේතය
E. ඇල්ගොරිතමය

17. “දත්ත කියවීම/පෙන්වීම සඳහා සමාන්තරාස්‍රය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. අත් අනුරේඛනය
B. ගැලීම් සටහන
C. ආදාන/ප්‍රතිදාන සංකේතය
D. සැකසුම් සංකේතය
E. සම්බන්ධකය

18. “පියවර පිළිවෙළ පෙන්වන ඊතලය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Trace table
B. තීරණ සංකේතය
C. ව්‍යාජ කේතය
D. ගැලීම් රේඛාව
E. වියළි ධාවනය

19. “ගැලීම් සටහනේ කොටස් යා කරන කුඩා කවය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. අන්ත සංකේතය
B. සම්බන්ධකය
C. ඇල්ගොරිතමය
D. ආදාන/ප්‍රතිදාන සංකේතය
E. අත් අනුරේඛනය

20. “පරිගණකයක් නැතිව අනුරේඛනයට තවත් නමක්” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ගැලීම් රේඛාව
B. Trace table
C. ගැලීම් සටහන
D. සැකසුම් සංකේතය
E. වියළි ධාවනය

ව්‍යුහගත ප්‍රශ්න

1. සංකේත පහ. *(ලකුණු 5)*

> අන්ත; සැකසුම; තීරණ; I/O; රේඛා.

2. විශාලතම pseudocode. *(ලකුණු 4)*

> largest; FOR; IF; DISPLAY.

3. ගැලීම් සටහන. *(ලකුණු 6)*

> Start; ලූපය; තීරණ; ප්‍රතිදානය.

4. trace table. *(ලකුණු 5)*

> පේළි 4; 45.

5. ගුණාංග තුනක්. *(ලකුණු 3)*

> සීමිත; පැහැදිලි; පිළිවෙළ.

සාරාංශය

- ඇල්ගොරිතමය සීමිත පැහැදිලි පියවර ය.
- ගැලීම් සටහන් සම්මත සංකේත භාවිත කරයි.
- ව්‍යාජ කේතය ව්‍යුහගත ඉංග්‍රීසි ය.
- Trace table අතින් අගයන් පරීක්ෂා කරයි.
- කේතනයට පෙර අනුරේඛනය කරන්න.

MCQ පිළිතුරු – 9.3

1. **B** – තීරණය.
2. **C** – I/O.
3. **D** – අන්තය.
4. **B** – ව්‍යාජ කේතය.
5. **E** – 45.
6. **E** – 45 පමණි.
7. **A** – Trace table.

8. **C** – සීමිත, පැහැදිලි.
9. **A** – ඇල්ගොරිතමය – ගැටලුවක් විසඳන සීමිත පිළිවෙළ පැහැදිලි පියවර.
10. **B** – ගැලීම් සටහන – සම්මත සංකේත භාවිත කරන ඇල්ගොරිතම රූපය.
11. **A** – ව්‍යාජ කේතය – ඇල්ගොරිතමයක ව්‍යුහගත ඉංග්‍රීසි විස්තරය.
12. **E** – අත් අනුරේඛනය – අගයන් පරීක්ෂාවට ඇල්ගොරිතමය අතින් ක්‍රියාත්මක කිරීම.
13. **D** – Trace table – අනුරේඛනයේදී විචල්‍ය අගයන් සටහන් කරන වගුව.
14. **D** – අන්ත සංකේතය – ආරම්භය හෝ නැවතුම පෙන්වන රවුම් කොටුව.
15. **C** – සැකසුම් සංකේතය – ගණනය හෝ පැවරුම සඳහා සෘජුකෝණාස්‍රය.
16. **A** – තීරණ සංකේතය – ඔව්/නැත කොන්දේසියක් සහිත දියමන්තිය.
17. **C** – ආදාන/ප්‍රතිදාන සංකේතය – දත්ත කියවීම/පෙන්වීම සඳහා සමාන්තරාස්‍රය.
18. **D** – ගැලීම් රේඛාව – පියවර පිළිවෙළ පෙන්වන ඊතලය.
19. **B** – සම්බන්ධකය – ගැලීම් සටහනේ කොටස් යා කරන කුඩා කවය.
20. **E** – වියළි ධාවනය – පරිගණකයක් නැතිව අනුරේඛනයට තවත් නමක්.

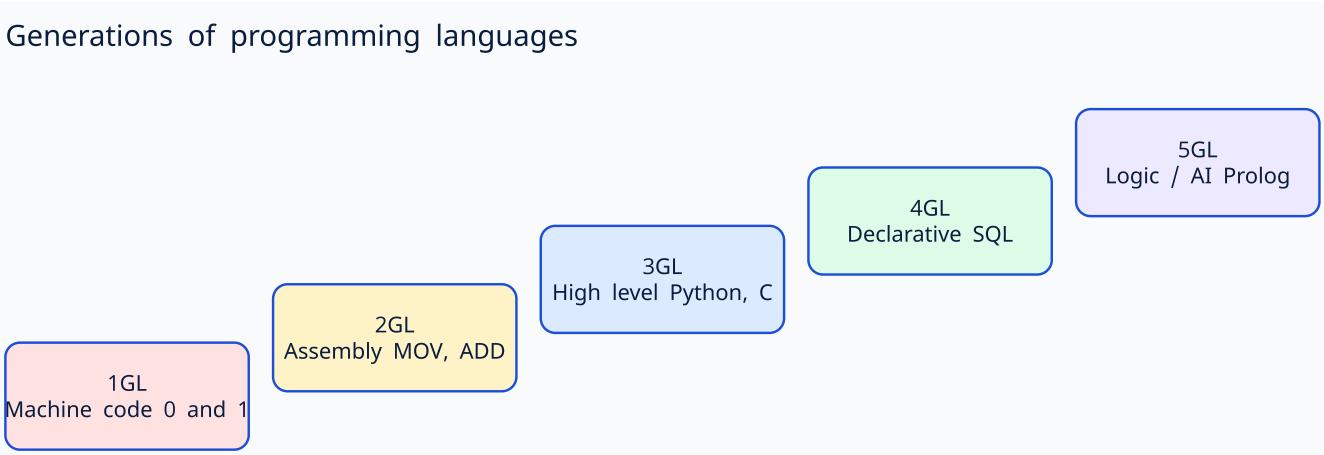
© 2026 schoolict.net (Ruwan Mangala Suraweera). සියලු හිමිකම් ඇවිරිණි.

13 ශ්‍රේණිය A/L ICT · පාඩම 9

9.4 ක්‍රමලේඛන භාෂා සහ ප්‍රතිමාන

විෂය නිර්දේශය 9.4. ක්‍රමලේඛන භාෂා යන්ත්‍ර කේතයේ සිට මිනිස් චින්තනයට සමීප භාෂා දක්වා **පරම්පරා** හරහා විකාශය විය. **ප්‍රතිමානයක්** යනු ක්‍රමලේඛන ශෛලියකි: **අනුඥාත්මක** (කෙසේද, පියවරෙන් පියවර), **ප්‍රකාශාත්මක** (අවශ්‍ය ප්‍රතිඵලය කුමක්ද) සහ **වස්තු නැඹුරු** (දත්ත සහ ක්‍රම එක් කරන වස්තු).

1. පරම්පරා



භාෂා පරම්පරා 1GL–5GL

පරම්පරාව	උදාහරණය	ලක්ෂණය
1GL යන්ත්‍ර	10110000 01100001	ද්විමය; සෘජුව ධාවනය; ලිවීම අපහසු
2GL assembly	MOV AL, 61h	mnemonics; assembler අවශ්‍යයි
3GL ඉහළ මට්ටම	Python, C, Java	ඉංග්‍රීසි වැනි; ගෙන යා හැකි
4GL	SQL	අවශ්‍ය දේ කියන්න
5GL	Prolog	AI සඳහා නීති සහ තර්කය

2. ප්‍රතිමාන

අනුඥාත්මක
විධාන පියවරෙන් පියවර තත්ත්වය වෙනස් කරයි.

ප්‍රකාශාත්මක
ප්‍රතිඵලය විස්තර කරයි; කෙසේද යන්න පද්ධතිය තීරණය කරයි.

වස්තු නැඹුරු
encapsulation, inheritance, polymorphism සහිත classes සහ objects.

🐍 එක් කාර්යයක් ශෛලී තුනකින්

```
prices = [120, 80, 200]

# imperative: say HOW, step by step
total = 0
for p in prices:
    total = total + p
print("Imperative:", total)

# declarative style: say WHAT you want
print("Declarative:", sum(prices))

# object oriented: data + methods in a class
class Cart:
    def __init__(self, items):
        self.items = items
    def total(self):
        return sum(self.items)

print("OOP:", Cart(prices).total())
```

▶ ප්‍රතිදානය

```
Imperative: 400
Declarative: 400
OOP: 400
```

🧱 **විසඳූ උදාහරණය: මිල 120, 80, 200. ශෛලී තුන එකඟද?**
අතින් එකතුව = **400**. අනුඥාත්මක ලූපය, ප්‍රකාශාත්මක sum() සහ Cart වස්තුව සියල්ල **400** මුද්‍රණය කළේය.

💡 Python බහු-ප්‍රතිමාන භාෂාවකි.

🚀 **විෂය නිර්දේශයෙන් ඔබ්බට (2026 යාවත්කාලීනය)** *විභාගයට අවශ්‍ය නැත – දැනුම සඳහා කියවන්න.* 2026 දී බොහෝ සංවර්ධකයින් AI වෙත ස්වභාවික භාෂාවෙන් අවශ්‍ය දේ කියයි; නමුත් කේතය කියවා පරීක්ෂා කරයි.

ප්‍රධාන පද

යන්ත්‍ර භාෂාව – CPU සෘජුව ධාවනය කරන පළමු පරම්පරාවේ ද්විමය කේතය.

Assembly භාෂාව – mnemonics භාවිත කරන දෙවන පරම්පරාවේ භාෂාව.

ඉහළ මට්ටමේ භාෂාව – ඉංග්‍රීසි වැනි ගෙන යා හැකි තෙවන පරම්පරාවේ භාෂාව.

හතරවන පරම්පරාවේ භාෂාව – SQL වැනි ප්‍රතිඵලය ප්‍රකාශ කරන භාෂාව.

පස්වන පරම්පරාවේ භාෂාව – Prolog වැනි AI සඳහා තර්ක පදනම් භාෂාව.

ක්‍රමලේඛන ප්‍රතිමානය – වැඩසටහන් ලිවීමේ ශෛලියක්.

අනුඥාත්මක ප්‍රතිමානය – ප්‍රකාශ පියවරෙන් පියවර තත්ත්වය වෙනස් කරන ශෛලිය.

ප්‍රකාශාත්මක ප්‍රතිමානය – කෙසේද නොව කුමක්ද යන්න ප්‍රකාශ කරන ශෛලිය.

වස්තු නැඹුරු ප්‍රතිමානය – දත්ත සහ ක්‍රම එක් කරන වස්තු මත පදනම් ශෛලිය.

Class – වස්තුවල උපලක්ෂණ සහ ක්‍රම නිර්වචනය කරන අච්චුව.

Object – තමන්ගේ දත්ත සහිත class එකක නිදසුන.

සංවෘතකරණය – දත්ත සහ ක්‍රම එකට තබා විස්තර සැඟවීම.

ආදර්ශ ප්‍රශ්න සහ පිළිතුරු

1. පරම්පරා පහ?

පිළිතුර: යන්ත්‍ර, assembly, ඉහළ, 4GL, 5GL.

2. 4GL උදාහරණයක්?

පිළිතුර: SQL.

3. අනුඥාත්මක සහ ප්‍රකාශාත්මක?

පිළිතුර: කෙසේද සහ කුමක්ද.

4. OOP ලක්ෂණ තුනක්?

පිළිතුර: encapsulation, inheritance, polymorphism.

5. ශෛලී තුනේ ප්‍රතිදානය?

පිළිතුර: 400.

6. Python බහු-ප්‍රතිමාන ඇයි?

පිළිතුර: ශෛලී කිහිපයකට සහාය දෙයි.

බහුවරණ ප්‍රශ්න

1. ද්විමය

A. SQL
B. Prolog
C. machine language
D. assembly language
E. Python

2. MOV

A. assembly language
B. Prolog
C. SQL
D. machine language
E. HTML

3. SQL

A. fifth generation
B. first generation
C. second generation
D. third generation
E. fourth generation

4. Prolog

A. a 3GL
B. a logic / 5GL language
C. assembly
D. a markup language
E. machine code

5. කුමක්ද

A. assembly
B. machine level
C. declarative
D. imperative
E. procedural

6. අච්චුව

A. compiler
B. loop
C. constant
D. class
E. tuple

7. ප්‍රතිදානය

A. 480
B. 120
C. 200
D. 400
E. 380

8. සැඟවීම

A. normalisation
B. encapsulation
C. compilation
D. iteration
E. recursion

9. “CPU සෘජුව ධාවනය කරන පළමු පරම්පරාවේ ද්විමය කේතය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ඉහළ මට්ටමේ භාෂාව
B. ප්‍රකාශාත්මක ප්‍රතිමානය
C. Object
D. ක්‍රමලේඛන ප්‍රතිමානය
E. යන්ත්‍ර භාෂාව

10. “mnemonics භාවිත කරන දෙවන පරම්පරාවේ භාෂාව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Assembly භාෂාව
B. අනුඥාත්මක ප්‍රතිමානය
C. සංවෘතකරණය
D. වස්තු නැඹුරු ප්‍රතිමානය
E. හතරවන පරම්පරාවේ භාෂාව

11. “ඉංග්‍රීසි වැනි ගෙන යා හැකි තෙවන පරම්පරාවේ භාෂාව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ප්‍රකාශාත්මක ප්‍රතිමානය
B. ඉහළ මට්ටමේ භාෂාව
C. පස්වන පරම්පරාවේ භාෂාව
D. යන්ත්‍ර භාෂාව
E. Class

12. “SQL වැනි ප්‍රතිඵලය ප්‍රකාශ කරන භාෂාව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. වස්තු නැඹුරු ප්‍රතිමානය
B. Assembly භාෂාව
C. හතරවන පරම්පරාවේ භාෂාව
D. Object
E. ක්‍රමලේඛන ප්‍රතිමානය

13. “Prolog වැනි AI සඳහා තර්ක පදනම් භාෂාව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Class
B. අනුඥාත්මක ප්‍රතිමානය
C. සංවෘතකරණය
D. ඉහළ මට්ටමේ භාෂාව
E. පස්වන පරම්පරාවේ භාෂාව

14. “වැඩසටහන් ලිවීමේ ශෛලියක්” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ක්‍රමලේඛන ප්‍රතිමානය
B. යන්ත්‍ර භාෂාව
C. Object
D. හතරවන පරම්පරාවේ භාෂාව
E. ප්‍රකාශාත්මක ප්‍රතිමානය

15. “ප්‍රකාශ පියවරෙන් පියවර තත්ත්වය වෙනස් කරන ශෛලිය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Assembly භාෂාව
B. සංවෘතකරණය
C. පස්වන පරම්පරාවේ භාෂාව
D. අනුඥාත්මක ප්‍රතිමානය
E. වස්තු නැඹුරු ප්‍රතිමානය

16. “කෙසේද නොව කුමක්ද යන්න ප්‍රකාශ කරන ශෛලිය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ක්‍රමලේඛන ප්‍රතිමානය
B. ඉහළ මට්ටමේ භාෂාව
C. යන්ත්‍ර භාෂාව
D. ප්‍රකාශාත්මක ප්‍රතිමානය
E. Class

17. “දත්ත සහ ක්‍රම එක් කරන වස්තු මත පදනම් ශෛලිය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. හතරවන පරම්පරාවේ භාෂාව
B. Object
C. අනුඥාත්මක ප්‍රතිමානය
D. Assembly භාෂාව
E. වස්තු නැඹුරු ප්‍රතිමානය

18. “වස්තුවල උපලක්ෂණ සහ ක්‍රම නිර්වචනය කරන අච්චුව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ඉහළ මට්ටමේ භාෂාව
B. Class
C. පස්වන පරම්පරාවේ භාෂාව
D. සංවෘතකරණය
E. ප්‍රකාශාත්මක ප්‍රතිමානය

19. “තමන්ගේ දත්ත සහිත class එකක නිදසුන” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ක්‍රමලේඛන ප්‍රතිමානය
B. හතරවන පරම්පරාවේ භාෂාව
C. Object
D. වස්තු නැඹුරු ප්‍රතිමානය
E. යන්ත්‍ර භාෂාව

20. “දත්ත සහ ක්‍රම එකට තබා විස්තර සැඟවීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. සංවෘතකරණය
B. Assembly භාෂාව
C. Class
D. පස්වන පරම්පරාවේ භාෂාව
E. අනුඥාත්මක ප්‍රතිමානය

ව්‍යුහගත ප්‍රශ්න

1. පරම්පරා පහ. *(ලකුණු 5)*

> 1GL; 2GL; 3GL; 4GL; 5GL.

2. ප්‍රතිමාන දෙක. *(ලකුණු 2)*

> කෙසේද; කුමක්ද.

3. class, object, encapsulation. *(ලකුණු 3)*

> අච්චුව; නිදසුන; එකතුව.

4. ඉහළ මට්ටමේ වාසි. *(ලකුණු 2)*

> පහසුය; ගෙන යා හැකිය.

5. ශෛලී දෙකකින් එකතුව. *(ලකුණු 2)*

> for ලූපය; sum().

සාරාංශය

- භාෂා 1GL සිට 5GL දක්වා විකාශය විය.
- ඉහළ පරම්පරා මිනිසුන්ට සමීපයි.
- අනුඥාත්මක කෙසේද කියයි.
- ප්‍රකාශාත්මක කුමක්ද කියයි.
- OOP classes සහ objects භාවිත කරයි.

MCQ පිළිතුරු – 9.4

1. **C** – 1GL.
2. **A** – 2GL.
3. **E** – 4GL.
4. **B** – 5GL.
5. **C** – ප්‍රකාශාත්මක.
6. **D** – Class.
7. **D** – 120 + 80 + 200.
8. **B** – Encapsulation.
9. **E** – යන්ත්‍ර භාෂාව – CPU සෘජුව ධාවනය කරන පළමු පරම්පරාවේ ද්විමය කේතය.
10. **A** – Assembly භාෂාව – mnemonics භාවිත කරන දෙවන පරම්පරාවේ භාෂාව.
11. **B** – ඉහළ මට්ටමේ භාෂාව – ඉංග්‍රීසි වැනි ගෙන යා හැකි තෙවන පරම්පරාවේ භාෂාව.
12. **C** – හතරවන පරම්පරාවේ භාෂාව – SQL වැනි ප්‍රතිඵලය ප්‍රකාශ කරන භාෂාව.
13. **E** – පස්වන පරම්පරාවේ භාෂාව – Prolog වැනි AI සඳහා තර්ක පදනම් භාෂාව.
14. **A** – ක්‍රමලේඛන ප්‍රතිමානය – වැඩසටහන් ලිවීමේ ශෛලියක්.
15. **D** – අනුඥාත්මක ප්‍රතිමානය – ප්‍රකාශ පියවරෙන් පියවර තත්ත්වය වෙනස් කරන ශෛලිය.
16. **D** – ප්‍රකාශාත්මක ප්‍රතිමානය – කෙසේද නොව කුමක්ද යන්න ප්‍රකාශ කරන ශෛලිය.
17. **E** – වස්තු නැඹුරු ප්‍රතිමානය – දත්ත සහ ක්‍රම එක් කරන වස්තු මත පදනම් ශෛලිය.
18. **B** – Class – වස්තුවල උපලක්ෂණ සහ ක්‍රම නිර්වචනය කරන අච්චුව.
19. **C** – Object – තමන්ගේ දත්ත සහිත class එකක නිදසුන.
20. **A** – සංවෘතකරණය – දත්ත සහ ක්‍රම එකට තබා විස්තර සැඟවීම.

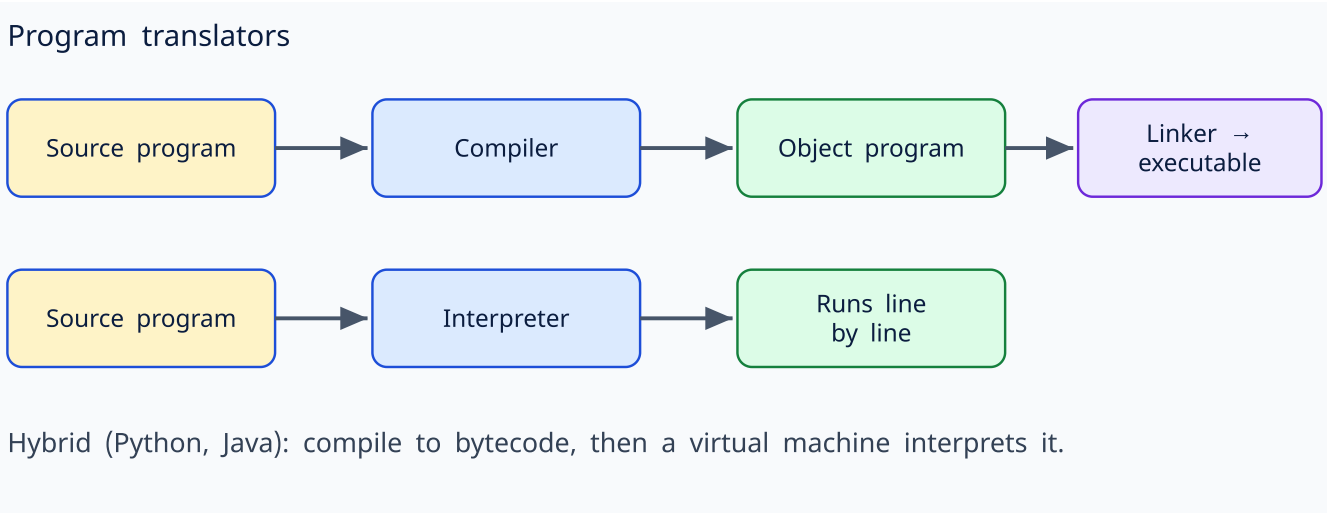
© 2026 schoolict.net (Ruwan Mangala Suraweera). සියලු හිමිකම් ඇවිරිණි.

13 ශ්‍රේණිය A/L ICT · පාඩම 9

9.5 වැඩසටහන් පරිවර්තනය: Compilers, Interpreters සහ Linkers

විෂය නිර්දේශය 9.5. CPU තේරුම් ගන්නේ යන්ත්‍ර කේතය පමණක් නිසා ඉහළ මට්ටමේ **මූලාශ්‍ර වැඩසටහනක් පරිවර්තනය** කළ යුතුය. **Compiler** සම්පූර්ණ වැඩසටහන **object program** එකක් බවට හරවයි; **interpreter** වරකට එක් ප්‍රකාශයක් පරිවර්තනය කර ධාවනය කරයි; **මිශ්‍ර ක්‍රමය** bytecode බවට compile කර virtual machine මගින් interpret කරයි. **Linker** object කේත සහ libraries එක් executable එකක් කරයි.

1. පරිවර්තක



Compiler, interpreter, linker සහ මිශ්‍ර

ලක්ෂණය	Compiler	Interpreter
පරිවර්තනය	සම්පූර්ණ වැඩසටහනම එකවර	වරකට එක් ප්‍රකාශයක්
ප්‍රතිදානය	Object/executable ගොනුව	වෙනම ගොනුවක් නැත
ධාවන වේගය	compile පසු වේගවත්	මන්දගාමී
දෝෂ	පරිවර්තනයෙන් පසු ලැයිස්තුව	පළමු දෝෂයේ නවතී
උදාහරණ	C, C++	JavaScript, BASIC

2. Python මිශ්‍ර ක්‍රමය භාවිත කරයි

🐍 ධාවනයට පෙර syntax දෝෂය සහ ධාවන දෝෂය

```
good = "x = 10\nprint(x * 2)"
bad = "print(10 +)"
code = compile(good, "good.py", "exec")   # translate to bytecode
exec(code)
try:
    compile(bad, "bad.py", "exec")
except SyntaxError:
    print("Syntax error: not translated")
try:
    print(10 / 0)
except ZeroDivisionError:
    print("Runtime error: division by zero")
```

▶ **ප්‍රතිදානය**

```
20
Syntax error: not translated
Runtime error: division by zero
```

🧱 **විසඳූ උදාහරණය: bad වැඩසටහන කිසි විටෙක ධාවනය නොවන්නේ ඇයි?**
compile() මුලින් **bytecode** බවට පරිවර්තනය කරයි. "print(10 +)" ව්‍යාකරණ දෝෂයක් නිසා **syntax error** එකක් ඇති වී කිසිවක් ධාවනය නොවේ. 10 / 0 ව්‍යාකරණ අතින් නිවැරදි නිසා පරිවර්තනය වේ; නමුත් **ධාවන කාලයේ** අසාර්ථක වේ.

💡 මූලාශ්‍ර = ඔබ ලියන දේ; object = පරිවර්තිත යන්ත්‍ර කේතය.

🚀 **විෂය නිර්දේශයෙන් ඔබ්බට (2026 යාවත්කාලීනය)** *විභාගයට අවශ්‍ය නැත – දැනුම සඳහා කියවන්න.* 2026 දී JavaScript engines සහ Python 3.13+ ධාවනය අතරතුර JIT compilation භාවිත කරයි.

ප්‍රධාන පද

මූලාශ්‍ර වැඩසටහන – ඉහළ මට්ටමේ භාෂාවෙන් ලියූ වැඩසටහන.

Object වැඩසටහන – මූලාශ්‍ර කේතය පරිවර්තනයෙන් ලැබෙන යන්ත්‍ර කේතය.

පරිවර්තකය – මූලාශ්‍ර කේතය යන්ත්‍ර කේතයට හරවන වැඩසටහන.

Compiler – ධාවනයට පෙර සම්පූර්ණ වැඩසටහන පරිවර්තනය කරයි.

Interpreter – වරකට එක් ප්‍රකාශයක් පරිවර්තනය කර ධාවනය කරයි.

මිශ්‍ර ක්‍රමය – bytecode වෙත compile කර VM මත interpret කිරීම.

Bytecode – virtual machine එකක් ධාවනය කරන අතරමැදි කේතය.

Linker – object ගොනු සහ libraries එක් executable එකක් කරයි.

Assembler – assembly භාෂාව යන්ත්‍ර කේතයට හරවයි.

Syntax දෝෂය – පරිවර්තනයේදී සොයාගන්නා භාෂා ව්‍යාකරණ දෝෂය.

ධාවන කාල දෝෂය – වැඩසටහන ධාවනය වන අතරතුර ඇතිවන දෝෂය.

Executable ගොනුව – මෙහෙයුම් පද්ධතියට සෘජුව ධාවනය කළ හැකි ගොනුව.

ආදර්ශ ප්‍රශ්න සහ පිළිතුරු

1. පරිවර්තනය අවශ්‍ය ඇයි?

පිළිතුර: CPU යන්ත්‍ර කේතය පමණක් තේරුම් ගනී.

2. Compiler සහ interpreter?

පිළිතුර: සම්පූර්ණයෙන්; එකින් එක.

3. මිශ්‍ර ක්‍රමය?

පිළිතුර: bytecode; VM.

4. Linker කරන්නේ?

පිළිතුර: object සහ libraries එකතු කරයි.

5. ධාවනය නොවන්නේ?

පිළිතුර: print(10 +).

6. මූලාශ්‍ර සහ object?

පිළිතුර: ලියූ; පරිවර්තිත.

බහුවරණ ප්‍රශ්න

1. සම්පූර්ණ

A. compiler
B. loader
C. interpreter
D. linker
E. editor

2. එකින් එක

A. interpreter
B. assembler
C. debugger
D. linker
E. compiler

3. libraries

A. compiler
B. linker
C. editor
D. interpreter
E. debugger

4. bytecode

A. pure compiler approach
B. assembly approach
C. linker approach
D. machine approach
E. hybrid approach

5. print(10 +)

A. logic error
B. syntax error
C. runtime error
D. no error
E. linker error

6. 10 / 0

A. runtime error
B. no error
C. syntax error
D. compile error
E. linker error

7. මියූ

A. bytecode only
B. source program
C. executable
D. object program
E. library

8. assembly

A. browser
B. interpreter
C. assembler
D. editor
E. linker

9. “ඉහළ මට්ටමේ භාෂාවෙන් ලියූ වැඩසටහන” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. පරිවර්තකය
B. Linker
C. මිශ්‍ර ක්‍රමය
D. මූලාශ්‍ර වැඩසටහන
E. ධාවන කාල දෝෂය

10. “මූලාශ්‍ර කේතය පරිවර්තනයෙන් ලැබෙන යන්ත්‍ර කේතය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Assembler
B. Bytecode
C. Object වැඩසටහන
D. Compiler
E. Executable ගොනුව

11. “මූලාශ්‍ර කේතය යන්ත්‍ර කේතයට හරවන වැඩසටහන” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. මූලාශ්‍ර වැඩසටහන
B. Linker
C. Interpreter
D. පරිවර්තකය
E. Syntax දෝෂය

12. “ධාවනයට පෙර සම්පූර්ණ වැඩසටහන පරිවර්තනය කරයි” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ධාවන කාල දෝෂය
B. Compiler
C. Object වැඩසටහන
D. මිශ්‍ර ක්‍රමය
E. Assembler

13. “වරකට එක් ප්‍රකාශයක් පරිවර්තනය කර ධාවනය කරයි” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Executable ගොනුව
B. Bytecode
C. පරිවර්තකය
D. Interpreter
E. Syntax දෝෂය

14. “bytecode වෙත compile කර VM මත interpret කිරීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ධාවන කාල දෝෂය
B. මූලාශ්‍ර වැඩසටහන
C. Compiler
D. Linker
E. මිශ්‍ර ක්‍රමය

15. “virtual machine එකක් ධාවනය කරන අතරමැදි කේතය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Object වැඩසටහන
B. Interpreter
C. Bytecode
D. Assembler
E. Executable ගොනුව

16. “object ගොනු සහ libraries එක් executable එකක් කරයි” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. මූලාශ්‍ර වැඩසටහන
B. මිශ්‍ර ක්‍රමය
C. Linker
D. Syntax දෝෂය
E. පරිවර්තකය

17. “assembly භාෂාව යන්ත්‍ර කේතයට හරවයි” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Bytecode
B. Object වැඩසටහන
C. Compiler
D. ධාවන කාල දෝෂය
E. Assembler

18. “පරිවර්තනයේදී සොයාගන්නා භාෂා ව්‍යාකරණ දෝෂය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Syntax දෝෂය
B. Executable ගොනුව
C. පරිවර්තකය
D. Interpreter
E. Linker

19. “වැඩසටහන ධාවනය වන අතරතුර ඇතිවන දෝෂය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Compiler
B. Assembler
C. මිශ්‍ර ක්‍රමය
D. මූලාශ්‍ර වැඩසටහන
E. ධාවන කාල දෝෂය

20. “මෙහෙයුම් පද්ධතියට සෘජුව ධාවනය කළ හැකි ගොනුව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Bytecode
B. Object වැඩසටහන
C. Syntax දෝෂය
D. Executable ගොනුව
E. Interpreter

ව්‍යුහගත ප්‍රශ්න

1. පරිවර්තනයේ අවශ්‍යතාව. *(ලකුණු 2)*

> යන්ත්‍ර කේතය පමණි; හැරවිය යුතුය.

2. Compiler සහ interpreter. *(ලකුණු 3)*

> සම්පූර්ණ/එකින්; ගොනුව; වේගය.

3. මිශ්‍ර ක්‍රමය. *(ලකුණු 3)*

> bytecode; VM; ගෙන යා හැකි.

4. Linker ශ්‍රිතය. *(ලකුණු 3)*

> එකතු කිරීම; executable.

5. Syntax සහ runtime. *(ලකුණු 2)*

> print(10 +); 10 / 0.

සාරාංශය

- මූලාශ්‍ර කේතය යන්ත්‍ර කේතයට හැරවිය යුතුය.
- Compiler සම්පූර්ණ වැඩසටහන හරවයි.
- Interpreter ප්‍රකාශයෙන් ප්‍රකාශය හරවයි.
- Python මිශ්‍රයි: bytecode + VM.
- Linker අවසන් executable සාදයි.

MCQ පිළිතුරු – 9.5

1. **A** – Compiler.
2. **A** – Interpreter.
3. **B** – Linker.
4. **E** – මිශ්‍ර.
5. **B** – ව්‍යාකරණ.
6. **A** – ධාවන කාලය.
7. **B** – මූලාශ්‍ර.
8. **C** – Assembler.
9. **D** – මූලාශ්‍ර වැඩසටහන – ඉහළ මට්ටමේ භාෂාවෙන් ලියූ වැඩසටහන.
10. **C** – Object වැඩසටහන – මූලාශ්‍ර කේතය පරිවර්තනයෙන් ලැබෙන යන්ත්‍ර කේතය.
11. **D** – පරිවර්තකය – මූලාශ්‍ර කේතය යන්ත්‍ර කේතයට හරවන වැඩසටහන.
12. **B** – Compiler – ධාවනයට පෙර සම්පූර්ණ වැඩසටහන පරිවර්තනය කරයි.
13. **D** – Interpreter – වරකට එක් ප්‍රකාශයක් පරිවර්තනය කර ධාවනය කරයි.
14. **E** – මිශ්‍ර ක්‍රමය – bytecode වෙත compile කර VM මත interpret කිරීම.
15. **C** – Bytecode – virtual machine එකක් ධාවනය කරන අතරමැදි කේතය.
16. **C** – Linker – object ගොනු සහ libraries එක් executable එකක් කරයි.
17. **E** – Assembler – assembly භාෂාව යන්ත්‍ර කේතයට හරවයි.
18. **A** – Syntax දෝෂය – පරිවර්තනයේදී සොයාගන්නා භාෂා ව්‍යාකරණ දෝෂය.
19. **E** – ධාවන කාල දෝෂය – වැඩසටහන ධාවනය වන අතරතුර ඇතිවන දෝෂය.
20. **D** – Executable ගොනුව – මෙහෙයුම් පද්ධතියට සෘජුව ධාවනය කළ හැකි ගොනුව.

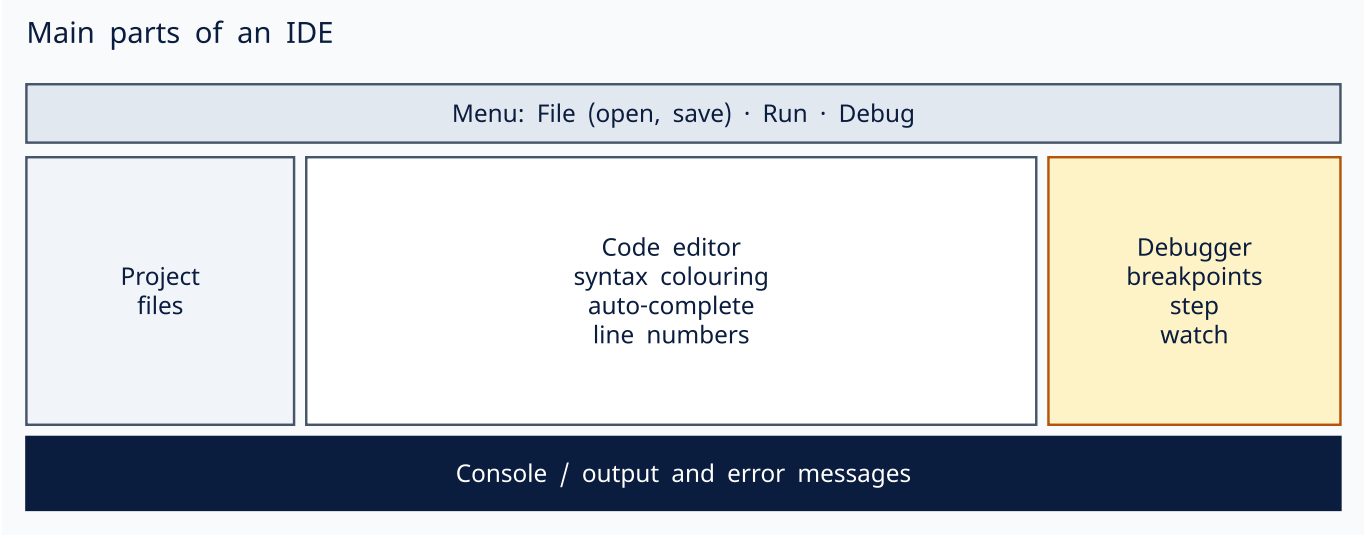
© 2026 schoolict.net (Ruwan Mangala Suraweera). සියලු හිමිකම් ඇවිරිණි.

13 ශ්‍රේණිය A/L ICT · පාඩම 9

9.6 ඒකාබද්ධ සංවර්ධන පරිසර (IDE) සහ Debug කිරීම

විෂය නිර්දේශය 9.6. IDE එකක් editor, පරිවර්තකය, ධාවකය සහ **debugger** එක් වැඩසටහනක එකතු කරයි (උදා: IDLE, Thonny, VS Code). **ගොනු විවෘත කිරීම සහ සුරැකීම**, වැඩසටහන් **ධාවනය** සහ breakpoints, stepping, variables watch වැනි **debug පහසුකම්** ඉගෙන ගනිමු.

1. IDE හි මූලික ලක්ෂණ



IDE හි කොටස්

- **Editor**: syntax වර්ණ, ස්වයං indent, auto-complete, පේළි අංක.
- **File මෙනුව**: New, Open, Save, Save As (.py ගොනු).
- **Run**: පරිවර්තනය සහ ධාවනය (IDLE හි F5); ප්‍රතිදානය **console** හි.
- **Debugger**: breakpoints, step into/over, විචල්‍ය watch, call stack.

2. off-by-one දෝෂයක් debug කිරීම

🐍 off-by-one දෝෂයක් debug කිරීම

```
marks = [60, 70, 80]

def average_bug(m):
    total = 0
    for i in range(len(m) - 1):   # bug: skips the last item
        total += m[i]
    return total / len(m)

def average_fixed(m):
    total = 0
    for i in range(len(m)):
        total += m[i]
    return total / len(m)

print("Buggy:", average_bug(marks))
print("Fixed:", average_fixed(marks))
```

▶ ප්‍රතිදානය

```
Buggy: 43.333333333333336
Fixed: 70.0
```

🧱 විසඳූ උදාහරණය: 60, 70, 80 හි සාමාන්‍යය වැරදියි. debugger හේතුව සොයන්නේ කෙසේද?

loop තුළ breakpoint තබා i සහ total watch කරන්න. i 0 සහ 1 පමණි; total 130; 130 ÷ 3 = **43.33**. range(len(m)) විය යුතුය; එවිට **70.0**.

දෝෂ වර්ගය	සොයන්නේ	උදාහරණය
Syntax	පරිවර්තකය, ධාවනයට පෙර	if පසු : නැත
ධාවන කාල	ධාවනයේදී බිඳවැටීම	ශූන්‍යයෙන් බෙදීම
තාර්කික	වැරදි ප්‍රතිදානය; debugger	range(len(m) - 1)

💡 තාර්කික දෝෂ පණිවිඩයක් නොදේ – බලාපොරොත්තු ප්‍රතිඵල සමඟ පරීක්ෂාව පමණක් ඒවා හෙළි කරයි.

🚀 **විෂය නිර්දේශයෙන් ඔබ්බට (2026 යාවත්කාලීනය)** *විභාගයට අවශ්‍ය නැත – දැනුම සඳහා කියවන්න.* 2026 IDE වල දෝෂ පැහැදිලි කර විසඳුම් යෝජනා කරන AI ඇත; පිළිගැනීමට පෙර පරීක්ෂා කරන්න.

ප්‍රධාන පද

IDE – editor, පරිවර්තකය, ධාවකය, debugger එකතු කළ මෘදුකාංගය.

කේත editor – මූලාශ්‍ර කේතය ටයිප් කරන IDE කොටස.

Syntax වර්ණ – කියවීමට පහසුවට keywords වර්ණ ගැන්වීම.

Auto-complete – ටයිප් කරන විට නම් යෝජනා කිරීම.

Console – ප්‍රතිදානය සහ දෝෂ පෙන්වන කවුළුව.

Debugger – දෝෂ සෙවීමට පියවරෙන් පියවර ධාවනය කරන මෙවලම.

Breakpoint – ධාවනය නවතින සලකුණු කළ පේළිය.

Step into – ඊළඟ පේළිය ධාවනය; ශ්‍රිතය ඇතුළට යාම.

Watch – තෝරාගත් විචල්‍යවල වත්මන් අගය පෙන්වීම.

තාර්කික දෝෂය – බිඳ නොවැටී වැරදි ප්‍රතිඵල දෙන වරද.

Off-by-one දෝෂය – එක් වරක් වැඩි හෝ අඩු වන ලූපය.

Call stack – දැනට ධාවනය වන ශ්‍රිත ලැයිස්තුව.

ආදර්ශ ප්‍රශ්න සහ පිළිතුරු

1. IDE ලක්ෂණ හතරක්.

> **පිළිතුර:** editor, ගොනු, run, debugger.

2. Breakpoint?

> **පිළිතුර:** ධාවනය නවතින පේළිය.

3. සාමාන්‍ය දෙක?

> **පිළිතුර:** 43.33 සහ 70.0.

4. දෝෂයට හේතුව?

> **පිළිතුර:** අවසන් අයිතමය මඟ හැරුණි.

5. පණිවිඩයක් නොදෙන දෝෂ?

> **පිළිතුර:** තාර්කික දෝෂ.

6. Python ගොනුව සුරැකීම?

> **පිළිතුර:** File → Save, .py.

බහුවරණ ප්‍රශ්න

1. නවතින

A. linker
B. breakpoint
C. console
D. comment
E. compiler

2. අගය පෙන්වීම

A. compile all
B. comment out
C. save as
D. link
E. watch

3. buggy

A. 80.00
B. 60.00
C. 70.00
D. 43.33
E. 65.00

4. fixed

A. 65.0
B. 210.0
C. 70.0
D. 80.0
E. 43.3

5. range

A. off-by-one logic error
B. linker error
C. syntax error
D. runtime crash
E. hardware error

6. ප්‍රතිදානය

A. structure chart
B. breakpoint
C. console
D. file menu
E. linker

7. : නැත

A. before the program runs (syntax error)
B. only after output
C. by the printer
D. by the linker
E. never

8. extension

A. .exe
B. .docx
C. .obj
D. .txt only
E. .py

9. “editor, පරිවර්තකය, ධාවකය, debugger එකතු කළ මෘදුකාංගය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Off-by-one දෝෂය
B. Step into
C. Debugger
D. IDE
E. Syntax වර්ණ

10. “මූලාශ්‍ර කේතය ටයිප් කරන IDE කොටස” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Auto-complete
B. Breakpoint
C. Watch
D. Call stack
E. කේත editor

11. “කියවීමට පහසුවට keywords වර්ණ ගැන්වීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Step into
B. Syntax වර්ණ
C. IDE
D. තාර්කික දෝෂය
E. Console

12. “ටයිප් කරන විට නම් යෝජනා කිරීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Auto-complete
B. Debugger
C. කේත editor
D. Watch
E. Off-by-one දෝෂය

13. “ප්‍රතිදානය සහ දෝෂ පෙන්වන කවුළුව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. තාර්කික දෝෂය
B. Console
C. Syntax වර්ණ
D. Breakpoint
E. Call stack

14. “දෝෂ සෙවීමට පියවරෙන් පියවර ධාවනය කරන මෙවලම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. IDE
B. Off-by-one දෝෂය
C. Debugger
D. Auto-complete
E. Step into

15. “ධාවනය නවතින සලකුණු කළ පේළිය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Call stack
B. Console
C. Watch
D. Breakpoint
E. කේත editor

16. “ඊළඟ පේළිය ධාවනය; ශ්‍රිතය ඇතුළට යාම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Step into
B. Debugger
C. Syntax වර්ණ
D. IDE
E. තාර්කික දෝෂය

17. “තෝරාගත් විචල්‍යවල වත්මන් අගය පෙන්වීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Off-by-one දෝෂය
B. Auto-complete
C. Watch
D. කේත editor
E. Breakpoint

18. “බිඳ නොවැටී වැරදි ප්‍රතිඵල දෙන වරද” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Step into
B. තාර්කික දෝෂය
C. Call stack
D. Syntax වර්ණ
E. Console

19. “එක් වරක් වැඩි හෝ අඩු වන ලූපය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Debugger
B. Auto-complete
C. Watch
D. Off-by-one දෝෂය
E. IDE

20. “දැනට ධාවනය වන ශ්‍රිත ලැයිස්තුව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Console
B. තාර්කික දෝෂය
C. කේත editor
D. Breakpoint
E. Call stack

ව්‍යුහගත ප්‍රශ්න

1. IDE ලක්ෂණ හතර. *(ලකුණු 4)*

> editor; ගොනු; run; debugger.

2. පියවර. *(ලකුණු 5)*

> Open; edit; Save; Run.

3. breakpoint, step, watch. *(ලකුණු 3)*

> නැවතීම; එක් පේළිය; අගය.

4. දෝෂය සොයා නිවැරදි කිරීම. *(ලකුණු 3)*

> range; 70.0.

5. දෝෂ වර්ග තුන. *(ලකුණු 3)*

> ව්‍යාකරණ; බිඳවැටීම; වැරදි ප්‍රතිදානය.

සාරාංශය

- IDE editor, ධාවකය, debugger එකතු කරයි.
- Python ගොනු .py ලෙස සුරකින්න.
- Breakpoints ධාවනය නවත්වයි.
- Step සහ watch තාර්කික දෝෂ හෙළි කරයි.
- තාර්කික දෝෂ සොයා ගැනීමට පරීක්ෂාව අවශ්‍යයි.

MCQ පිළිතුරු – 9.6

1. **B** – Breakpoint.
2. **E** – Watch.
3. **D** – 130 ÷ 3.
4. **C** – 210 ÷ 3.
5. **A** – තාර්කික.
6. **C** – Console.
7. **A** – Syntax.

8. **E** – .py.
9. **D** – IDE – editor, පරිවර්තකය, ධාවකය, debugger එකතු කළ මෘදුකාංගය.
10. **E** – කේත editor – මූලාශ්‍ර කේතය ටයිප් කරන IDE කොටස.
11. **B** – Syntax වර්ණ – කියවීමට පහසුවට keywords වර්ණ ගැන්වීම.
12. **A** – Auto-complete – ටයිප් කරන විට නම් යෝජනා කිරීම.
13. **B** – Console – ප්‍රතිදානය සහ දෝෂ පෙන්වන කවුළුව.
14. **C** – Debugger – දෝෂ සෙවීමට පියවරෙන් පියවර ධාවනය කරන මෙවලම.
15. **D** – Breakpoint – ධාවනය නවතින සලකුණු කළ පේළිය.
16. **A** – Step into – ඊළඟ පේළිය ධාවනය; ශ්‍රිතය ඇතුළට යාම.
17. **C** – Watch – තෝරාගත් විචල්‍යවල වත්මන් අගය පෙන්වීම.
18. **B** – තාර්කික දෝෂය – බිඳ නොවැටී වැරදි ප්‍රතිඵල දෙන වරද.
19. **D** – Off-by-one දෝෂය – එක් වරක් වැඩි හෝ අඩු වන ලූපය.
20. **E** – Call stack – දැනට ධාවනය වන ශ්‍රිත ලැයිස්තුව.

© 2026 schoolict.net (Ruwan Mangala Suraweera). සියලු හිමිකම් ඇවිරිණි.

13 ශ්‍රේණිය A/L ICT · පාඩම 9

9.7 Python මූලික: ව්‍යුහය, විචල්‍ය, දත්ත වර්ග, කාරක සහ ආදාන/ප්‍රතිදාන

විෂය නිර්දේශය 9.7. ඇල්ගොරිතම කේතනයට අනුඥාත්මක භාෂාවක් – Python – භාවිත කරමු. වැඩසටහනක **ව්‍යුහය**, **විවරණ**, **නියත සහ විචල්‍ය**, **ප්‍රාථමික දත්ත වර්ග**, **අංක ගණිත, සම්බන්ධතා, තාර්කික සහ bitwise කාරක**, **කාරක ප්‍රමුඛතාව**, යතුරුපුවරු **ආදානය** සහ තිර **ප්‍රතිදානය** හඳුනා ගනිමු.

1. ව්‍යුහය, විවරණ, නියත සහ විචල්‍ය

🐍 නියත, විචල්‍ය, වර්ග සහ ආදානය

```
RATE = 0.08                      # constant (by convention UPPER CASE)
name = input("Name: ")           # str
qty = int(input("Quantity: "))   # int
price = 250.0                    # float
paid = True                      # bool
total = qty * price * (1 + RATE)
print(name, "pays Rs.", round(total, 2))
print(type(qty).__name__, type(price).__name__, type(paid).__name__)
```

▶ ප්‍රතිදානය

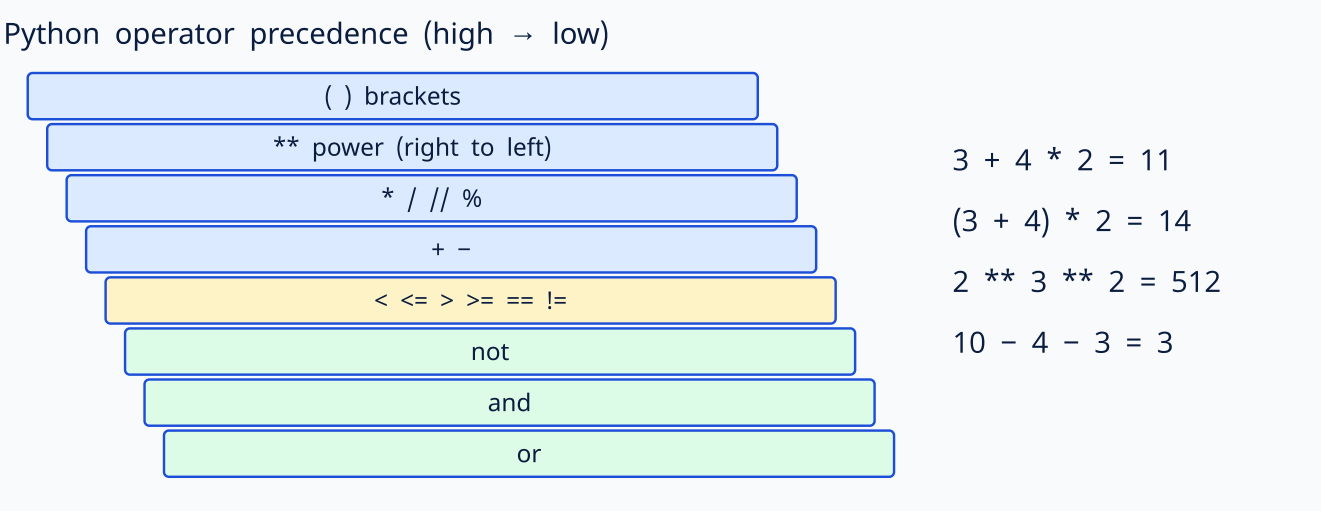
```
Name: Nimal
Quantity: 4
Nimal pays Rs. 1080.0
int float bool
```

වර්ගය	උදාහරණය	අර්ථය
int	4	පූර්ණ සංඛ්‍යාව
float	250.0	දශම සංඛ්‍යාව
str	"Nimal"	පෙළ
bool	True / False	සත්‍ය අගය

විවරණය # වලින් ආරම්භ වී පරිවර්තකය නොසලකයි. **විචල්‍යය** අගය වෙනස් විය හැකි නම් කළ මතක ස්ථානයකි; **නියතය** (සම්ප්‍රදායට UPPER CASE) වෙනස් නොවිය යුතුය. input() සැමවිටම **str** ලබා දෙන නිසා int() හෝ float() මගින් හරවයි.

> 🧮 **විසඳූ උදාහරණය: Nimal Rs. 250.0 බැගින් 4ක් 8% බදු සමඟ.**
> 4 × 250.0 = 1000.0; × 1.08 = **1080.0**. type() int, float, bool පෙන්වයි.

2. කාරක සහ ප්‍රමුඛතාව



කාරක ප්‍රමුඛතාව

කාණ්ඩය	කාරක	උදාහරණය → ප්‍රතිඵලය
අංක ගණිත	+ - * / // % **	7 // 2 → 3, 7 % 2 → 1, 7 / 2 → 3.5
සම්බන්ධතා	== != < <= > >=	5 > 3 → True
තාර්කික	and or not	5 > 3 and 2 > 8 → False
Bitwise	& | ^ ~ << >>	5 & 3 → 1, 5 | 3 → 7, 5 ^ 3 → 6, 5 << 1 → 10

🐍 කාරක සහ ප්‍රමුඛතාව

```
print(7 // 2, 7 % 2, 7 / 2)     # arithmetic
print(3 + 4 * 2, (3 + 4) * 2)  # precedence
print(2 ** 3 ** 2)             # ** is right to left
print(10 - 4 - 3)              # - is left to right
print(5 > 3 and 2 > 8)         # relational + logical
print(not True or True)
print(5 & 3, 5 | 3, 5 ^ 3, 5 << 1)   # bitwise
```

▶ ප්‍රතිදානය

```
3 1 3.5
11 14
512
3
False
True
1 7 6 10
```

🧮 **විසඳූ උදාහරණය: 2 3 2 සහ 5 ^ 3 පැහැදිලි කරන්න.**
දකුණේ සිට වමට: **3** 2 = 9, 2 **9** = 512. **101 XOR 011 = 110** = 6**.

💡 Python හි ^ යනු XOR ය; බලයට ** භාවිත කරන්න.

🚀 විෂය නිර්දේශයෙන් ඔබ්බට (2026 යාවත්කාලීනය) *විභාගයට අවශ්‍ය නැත – දැනුම සඳහා කියවන්න.* 2026 Python 3.12+ පැහැදිලි දෝෂ පණිවිඩ සහ IDE පරීක්ෂා කරන type hints (qty: int) ලබා දෙයි.

ප්‍රධාන පද

විවරණය – ධාවනය නොවන # වලින් ආරම්භ වන සටහන.

විචල්‍යය – අගය වෙනස් විය හැකි නම් කළ ගබඩා ස්ථානය.

නියතය – ධාවනයේදී වෙනස් නොවිය යුතු නම් කළ අගය.

ප්‍රාථමික දත්ත වර්ගය – int, float, str, bool වැනි මූලික වර්ගය.

පූර්ණ සංඛ්‍යාව – int පූර්ණ සංඛ්‍යා වර්ගය.

Float – දශම කොටසක් ඇති සංඛ්‍යා වර්ගය.

Boolean – True හෝ False රඳවන වර්ගය.

අංක ගණිත කාරකය – සංඛ්‍යා ගණනය කරන + හෝ // වැනි කාරකය.

සම්බන්ධතා කාරකය – අගයන් සසඳා True/False දෙන කාරකය.

තාර්කික කාරකය – කොන්දේසි එක් කරන and, or, not.

Bitwise කාරකය – & හෝ ^ වැනි bits මත ක්‍රියා කරන කාරකය.

කාරක ප්‍රමුඛතාව – කාරක ගණනය කරන පිළිවෙළ.

ආදර්ශ ප්‍රශ්න සහ පිළිතුරු

1. input() ලබා දෙන්නේ?

පිළිතුර: str.

2. 7 // 2 සහ 7 % 2?

පිළිතුර: 3 සහ 1.

3. 3 + 4 * 2?

පිළිතුර: 11.

4. 10 - 4 - 3?

පිළිතුර: 3.

5. ප්‍රාථමික වර්ග හතර?

පිළිතුර: int, float, str, bool.

6. නියතය සහ විචල්‍යය?

> **පිළිතුර:** නියතය වෙනස් නොවේ.

බහුවරණ ප්‍රශ්න

1. 7 // 2

A. 1
B. 3.5
C. 4
D. 3
E. 2

2. 7 % 2

A. 1
B. 0
C. 2
D. 3.5
E. 3

3. 3 + 4 * 2

A. 14
B. 11
C. 9
D. 10
E. 24

4. 2 3 2

A. 64
B. 12
C. 512
D. 256
E. 36

5. 5 ^ 3

A. 6
B. 2
C. 8
D. 125
E. 15

6. not True or True

A. 0
B. error
C. False
D. True
E. None

7. input()

A. bool
B. float
C. int
D. list
E. str

8. එකතුව

A. 1008.0
B. 1000.0
C. 1080.0
D. 1250.0
E. 1800.0

9. “ධාවනය නොවන # වලින් ආරම්භ වන සටහන” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Bitwise කාරකය
B. විවරණය
C. අංක ගණිත කාරකය
D. Float
E. නියතය

10. “අගය වෙනස් විය හැකි නම් කළ ගබඩා ස්ථානය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. කාරක ප්‍රමුඛතාව
B. සම්බන්ධතා කාරකය
C. විචල්‍යය
D. ප්‍රාථමික දත්ත වර්ගය
E. Boolean

11. “ධාවනයේදී වෙනස් නොවිය යුතු නම් කළ අගය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. පූර්ණ සංඛ්‍යාව
B. තාර්කික කාරකය
C. අංක ගණිත කාරකය
D. නියතය
E. විවරණය

12. “int, float, str, bool වැනි මූලික වර්ගය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Bitwise කාරකය
B. විචල්‍යය
C. සම්බන්ධතා කාරකය
D. Float
E. ප්‍රාථමික දත්ත වර්ගය

13. “int පූර්ණ සංඛ්‍යා වර්ගය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. තාර්කික කාරකය
B. පූර්ණ සංඛ්‍යාව
C. කාරක ප්‍රමුඛතාව
D. Boolean
E. නියතය

14. “දශම කොටසක් ඇති සංඛ්‍යා වර්ගය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Bitwise කාරකය
B. විවරණය
C. ප්‍රාථමික දත්ත වර්ගය
D. අංක ගණිත කාරකය
E. Float

15. “True හෝ False රඳවන වර්ගය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. කාරක ප්‍රමුඛතාව
B. Boolean
C. සම්බන්ධතා කාරකය
D. පූර්ණ සංඛ්‍යාව
E. විචල්‍යය

16. “සංඛ්‍යා ගණනය කරන + හෝ // වැනි කාරකය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. අංක ගණිත කාරකය
B. නියතය
C. තාර්කික කාරකය
D. විවරණය
E. Float

17. “අගයන් සසඳා True/False දෙන කාරකය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Bitwise කාරකය
B. Boolean
C. ප්‍රාථමික දත්ත වර්ගය
D. සම්බන්ධතා කාරකය
E. විචල්‍යය

18. “කොන්දේසි එක් කරන and, or, not” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. තාර්කික කාරකය
B. නියතය
C. පූර්ණ සංඛ්‍යාව
D. කාරක ප්‍රමුඛතාව
E. අංක ගණිත කාරකය

19. “& හෝ ^ වැනි bits මත ක්‍රියා කරන කාරකය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. විවරණය
B. Float
C. Bitwise කාරකය
D. සම්බන්ධතා කාරකය
E. ප්‍රාථමික දත්ත වර්ගය

20. “කාරක ගණනය කරන පිළිවෙළ” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. තාර්කික කාරකය
B. පූර්ණ සංඛ්‍යාව
C. විචල්‍යය
D. Boolean
E. කාරක ප්‍රමුඛතාව

ව්‍යුහගත ප්‍රශ්න

1. ප්‍රාථමික වර්ග. *(ලකුණු 4)*

> int; float; str; bool.

2. විචල්‍ය, නියත, විවරණ. *(ලකුණු 3)*

> qty; RATE; #.

3. ගණනය කරන්න. *(ලකුණු 3)*

> 11; 14; 512.

4. කාරක තුන. *(ලකුණු 3)*

> True; False; 1.

5. 8% බදු වැඩසටහන. *(ලකුණු 3)*

> input; ගණනය; print.

සාරාංශය

- විවරණ # වලින් ආරම්භ වේ.

- int, float, str, bool ප්‍රාථමික වර්ග ය.
- input() str දෙයි – හරවන්න.
- ** පසු * / // %, පසු + −.
- ^ XOR; ** බලය.

MCQ පිළිතුරු – 9.7

1. **D** – floor බෙදීම.
2. **A** – ශේෂය.
3. **B** – * මුලින්.
4. **C** – දකුණේ සිට.
5. **A** – XOR.
6. **D** – not මුලින්.
7. **E** – සැමවිටම str.
8. **C** – 1000 × 1.08.
9. **B** – විවරණය – ධාවනය නොවන # වලින් ආරම්භ වන සටහන.
10. **C** – විචල්‍යය – අගය වෙනස් විය හැකි නම් කළ ගබඩා ස්ථානය.
11. **D** – නියතය – ධාවනයේදී වෙනස් නොවිය යුතු නම් කළ අගය.
12. **E** – ප්‍රාථමික දත්ත වර්ගය – int, float, str, bool වැනි මූලික වර්ගය.
13. **B** – පූර්ණ සංඛ්‍යාව – int පූර්ණ සංඛ්‍යා වර්ගය.
14. **E** – Float – දශම කොටසක් ඇති සංඛ්‍යා වර්ගය.
15. **B** – Boolean – True හෝ False රඳවන වර්ගය.
16. **A** – අංක ගණිත කාරකය – සංඛ්‍යා ගණනය කරන + හෝ // වැනි කාරකය.
17. **D** – සම්බන්ධතා කාරකය – අගයන් සසඳා True/False දෙන කාරකය.
18. **A** – තාර්කික කාරකය – කොන්දේසි එක් කරන and, or, not.
19. **C** – Bitwise කාරකය – & හෝ ^ වැනි bits මත ක්‍රියා කරන කාරකය.
20. **E** – කාරක ප්‍රමුඛතාව – කාරක ගණනය කරන පිළිවෙළ.

© 2026 schoolict.net (Ruwan Mangala Suraweera). සියලු හිමිකම් ඇවිරිණි.

13 ශ්‍රේණිය A/L ICT · පාඩම 9

9.8 පාලන ව්‍යුහ: අනුක්‍රමය, වරණය සහ පුනරාවර්තනය

විෂය නිර්දේශය 9.8. සෑම ඇල්ගොරිතමයක්ම **පාලන ව්‍යුහ** තුනකින් ගොඩනැගේ: **අනුක්‍රමය** (පිළිවෙළට පියවර), **වරණය** (if / elif / else) සහ **පුනරාවර්තනය** – දන්නා වාර ගණනක් සඳහා **iteration** (for) සහ කොන්දේසියක් සත්‍ය වන තුරු **looping** (while). ව්‍යුහ එකක් තුළ තවත් එකක් **කූඩු** කළ හැක.

1. ව්‍යුහ තුන

අනුක්‍රමය
ප්‍රකාශ එකින් එක ධාවනය වේ.

වරණය
කොන්දේසියක් මගින් මාර්ගය තෝරයි.

පුනරාවර්තනය
for නියමිත වාර ගණනක්; while කොන්දේසිය සත්‍ය තුරු.

🐍 අනුක්‍රමය, වරණය සහ පුනරාවර්තනය

```
mark = 67
if mark >= 75:
    print("A")
elif mark >= 65:
    print("B")
elif mark >= 50:
    print("C")
else:
    print("F")

for i in range(1, 4):        # iteration (count-controlled)
    print(i, end=" ")
print()

n = 472                      # looping (condition-controlled)
digit_sum = 0
while n > 0:
    digit_sum += n % 10
    n = n // 10
print("Digit sum:", digit_sum)

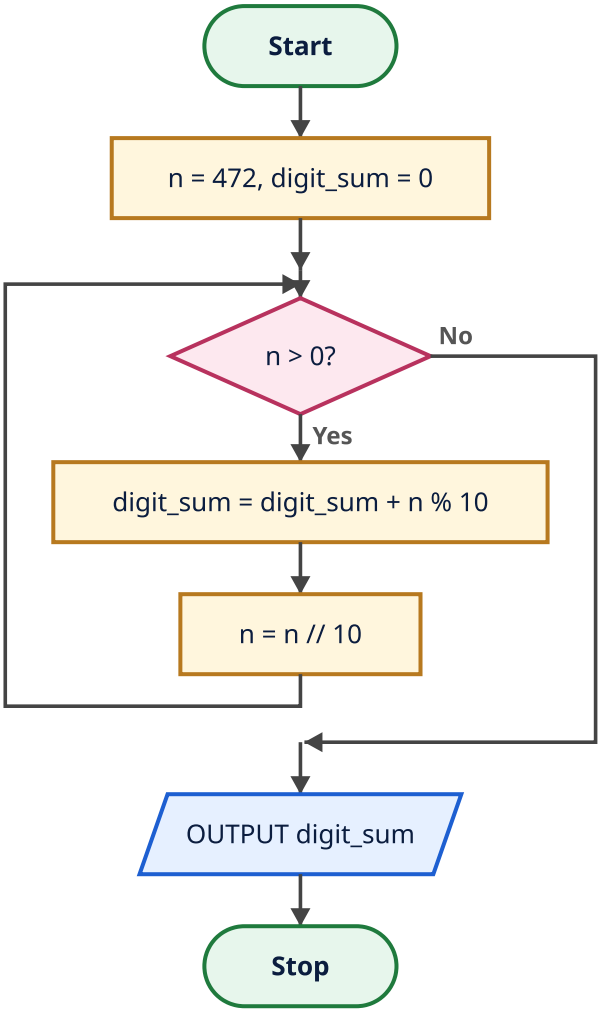
for r in range(1, 4):        # nested loops
    for c in range(1, 4):
        print(r * c, end=" ")
    print()
```

▶ ප්‍රතිදානය

```
B
1 2 3
Digit sum: 13
1 2 3
2 4 6
3 6 9
```

🧱 **විසඳූ උදාහරණය: ලකුණ 67 B මුද්‍රණය කරන්නේ ඇයි?**
කොන්දේසි පිළිවෙළට: 67 ≥ 75 False, 67 ≥ 65 **True**, එබැවින් B; ඉතිරිය මඟ හැරේ.

2. while ලූපය අනුරේඛනය



ගැලීම් සටහන: 472 හි ඉලක්කම් එකතුව

n % 10	digit_sum	// 10 පසු n
2	2	47
7	9	4
4	13	0

🧮 **විසඳූ උදාහරණය: 472 සඳහා අනුරේඛනය.**
ලූපය **3** වරක් ධාවනය වේ; n 0 වූ විට නවතී; **13** මුද්‍රණය කරයි.

💡 වාර ගණන දන්නේ නම් for; කොන්දේසියකින් නවතින්නේ නම් while. indent කිරීම ඇතුළත ඇති දේ පෙන්වයි.

🚀 **විෂය නිර්දේශයෙන් ඔබ්බට (2026 යාවත්කාලීනය)** *විභාගයට අවශ්‍ය නැත – දැනුම සඳහා කියවන්න.* Python 3.10+ match-case එක් කරයි; දිගු elif දාම වෙනුවට 2026 බොහෝ වැඩසටහන් භාවිත කරයි.

ප්‍රධාන පද

පාලන ව්‍යුහය – ප්‍රකාශ ධාවනය වන පිළිවෙළ පාලනය කරන ක්‍රමය.

අනුක්‍රමය – එකින් එක ධාවනය වන ප්‍රකාශ.

වරණය – කොන්දේසියක් අනුව මාර්ගයක් තේරීම.

පුනරාවර්තනය – ප්‍රකාශ කොටසක් වරකට වඩා ධාවනය.

Iteration – for වැනි ගණන් පාලිත පුනරාවර්තනය.

Looping – while වැනි කොන්දේසි පාලිත පුනරාවර්තනය.

if ප්‍රකාශය – කොන්දේසිය සත්‍ය නම් පමණක් ධාවනය.

elif – පෙර ඒවා අසත්‍ය වූ විට පරීක්ෂා කරන තවත් කොන්දේසිය.

for ලූපය – පරාසයක සෑම අගයකටම පුනරාවර්තනය.

while ලූපය – කොන්දේසිය සත්‍ය තුරු පුනරාවර්තනය.

කූඩු ව්‍යුහය – තවත් එකක් තුළ ඇති පාලන ව්‍යුහය.

අනන්ත ලූපය – කොන්දේසිය කිසි විටෙක අසත්‍ය නොවන ලූපය.

ආදර්ශ ප්‍රශ්න සහ පිළිතුරු

1. පාලන ව්‍යුහ තුන?

පිළිතුර: අනුක්‍රමය, වරණය, පුනරාවර්තනය.

2. Iteration සහ looping?

පිළිතුර: for; while.

3. range(1, 4)?

පිළිතුර: 1, 2, 3.

4. 472 ඉලක්කම් එකතුව?

> **පිළිතුර:** 13.

5. 3 × 3 අවසන් පේළිය?

> **පිළිතුර:** 3 6 9.

6. අනන්ත ලූපය?

> **පිළිතුර:** කොන්දේසිය අසත්‍ය නොවීම.

බහුවරණ ප්‍රශ්න

1. if

A. iteration
B. recursion
C. looping
D. selection
E. sequence

2. for

A. iteration (count-controlled)
B. sequence only
C. selection
D. looping (condition-controlled)
E. a function

3. 67

A. nothing
B. A
C. F
D. B
E. C

4. range(1, 4)

A. 4 3 2 1
B. 0 1 2 3
C. 1 2 3 4
D. 1 4
E. 1 2 3

5. 472

A. 14
B. 472
C. 7
D. 11
E. 13

6. වාර

A. 472 times
B. 2 times
C. 1 time
D. 3 times
E. 4 times

7. කූඩු

A. compiling
B. nesting
C. sequencing
D. linking
E. tracing

8. මැද පේළිය

A. 1 2 3
B. 4 5 6
C. 3 6 9
D. 2 4 6
E. 2 3 4

9. “ප්‍රකාශ ධාවනය වන පිළිවෙළ පාලනය කරන ක්‍රමය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Looping
B. වරණය
C. පාලන ව්‍යුහය
D. කූඩු ව්‍යුහය
E. elif

10. “එකින් එක ධාවනය වන ප්‍රකාශ” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. අනන්ත ලූපය
B. අනුක්‍රමය
C. පුනරාවර්තනය
D. for ලූපය
E. if ප්‍රකාශය

11. “කොන්දේසියක් අනුව මාර්ගයක් තේරීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. while ලූපය
B. elif
C. වරණය
D. Iteration
E. පාලන ව්‍යුහය

12. “ප්‍රකාශ කොටසක් වරකට වඩා ධාවනය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. පුනරාවර්තනය
B. කූඩු ව්‍යුහය
C. අනුක්‍රමය
D. Looping
E. for ලූපය

13. “for වැනි ගණන් පාලිත පුනරාවර්තනය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. වරණය
B. Iteration
C. while ලූපය
D. if ප්‍රකාශය
E. අනන්ත ලූපය

14. “while වැනි කොන්දේසි පාලිත පුනරාවර්තනය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Looping
B. පුනරාවර්තනය
C. elif
D. පාලන ව්‍යුහය
E. කූඩු ව්‍යුහය

15. “කොන්දේසිය සත්‍ය නම් පමණක් ධාවනය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. අනන්ත ලූපය
B. for ලූපය
C. if ප්‍රකාශය
D. අනුක්‍රමය
E. Iteration

16. “පෙර ඒවා අසත්‍ය වූ විට පරීක්ෂා කරන තවත් කොන්දේසිය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. while ලූපය
B. වරණය
C. පාලන ව්‍යුහය
D. Looping
E. elif

17. “පරාසයක සෑම අගයකටම පුනරාවර්තනය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. අනුක්‍රමය
B. කූඩු ව්‍යුහය
C. for ලූපය
D. if ප්‍රකාශය
E. පුනරාවර්තනය

18. “කොන්දේසිය සත්‍ය තුරු පුනරාවර්තනය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. elif
B. while ලූපය
C. වරණය
D. අනන්ත ලූපය
E. Iteration

19. “තවත් එකක් තුළ ඇති පාලන ව්‍යුහය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. කූඩු ව්‍යුහය
B. පාලන ව්‍යුහය
C. Looping
D. පුනරාවර්තනය
E. for ලූපය

20. “කොන්දේසිය කිසි විටෙක අසත්‍ය නොවන ලූපය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. අනුක්‍රමය
B. if ප්‍රකාශය
C. while ලූපය
D. Iteration
E. අනන්ත ලූපය

ව්‍යුහගත ප්‍රශ්න

1. ව්‍යුහ තුන. *(ලකුණු 3)*

> අනුක්‍රමය; වරණය; පුනරාවර්තනය.

2. ශ්‍රේණි වැඩසටහන. *(ලකුණු 4)*

> if; elif; else.

3. අනුරේඛනය. *(ලකුණු 4)*

> පේළි 3; 13.

4. for සහ while. *(ලකුණු 3)*

> ගණන; කොන්දේසිය.

5. කූඩු ලූප. *(ලකුණු 4)*

```
for; for; print.
```

සාරාංශය

- අනුක්‍රමය, වරණය, පුනරාවර්තනය.
- if/elif/else පිළිවෙළට පරීක්ෂා කරයි.
- for = දන්නා වාර.
- while = කොන්දේසිය අසත්‍ය තුරු.
- ව්‍යුහ කූඩු කළ හැක.

MCQ පිළිතුරු – 9.8

1. **D** – වරණය.
2. **A** – Iteration.
3. **D** – 67 ≥ 65.
4. **E** – 4 ඇතුළත් නොවේ.
5. **E** – 4 + 7 + 2.
6. **D** – ඉලක්කම් 3.
7. **B** – කූඩු.
8. **D** – 2 × 1..3.
9. **C** – පාලන ව්‍යුහය – ප්‍රකාශ ධාවනය වන පිළිවෙළ පාලනය කරන ක්‍රමය.
10. **B** – අනුක්‍රමය – එකින් එක ධාවනය වන ප්‍රකාශ.
11. **C** – වරණය – කොන්දේසියක් අනුව මාර්ගයක් තේරීම.
12. **A** – පුනරාවර්තනය – ප්‍රකාශ කොටසක් වරකට වඩා ධාවනය.
13. **B** – Iteration – for වැනි ගණන් පාලිත පුනරාවර්තනය.
14. **A** – Looping – while වැනි කොන්දේසි පාලිත පුනරාවර්තනය.
15. **C** – if ප්‍රකාශය – කොන්දේසිය සත්‍ය නම් පමණක් ධාවනය.
16. **E** – elif – පෙර ඒවා අසත්‍ය වූ විට පරීක්ෂා කරන තවත් කොන්දේසිය.
17. **C** – for ලූපය – පරාසයක සෑම අගයකටම පුනරාවර්තනය.
18. **B** – while ලූපය – කොන්දේසිය සත්‍ය තුරු පුනරාවර්තනය.
19. **A** – කූඩු ව්‍යුහය – තවත් එකක් තුළ ඇති පාලන ව්‍යුහය.
20. **E** – අනන්ත ලූපය – කොන්දේසිය කිසි විටෙක අසත්‍ය නොවන ලූපය.

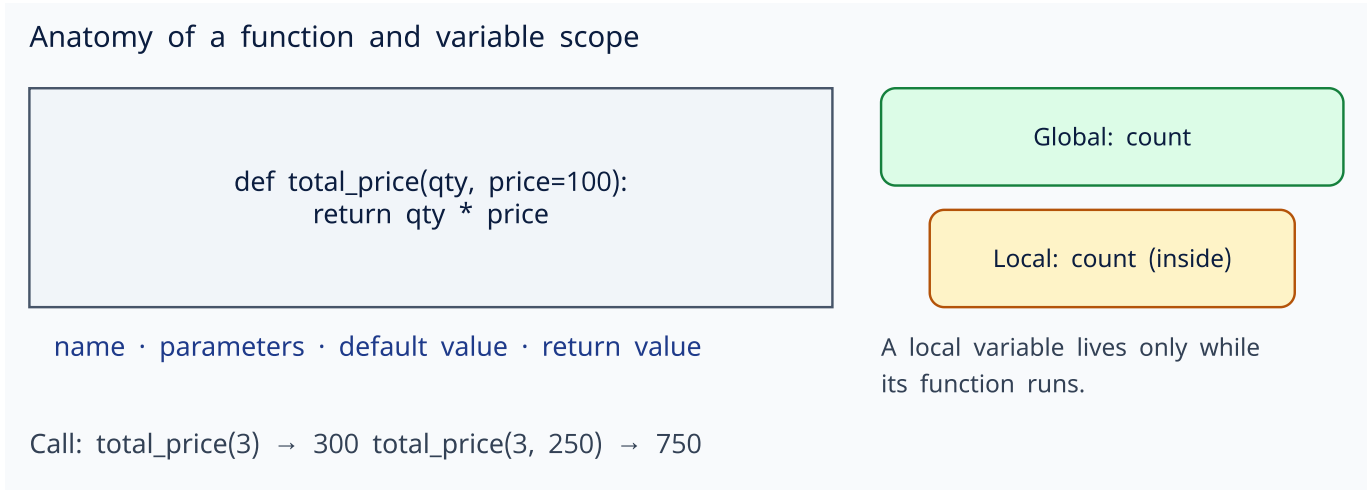
© 2026 schoolict.net (Ruwan Mangala Suraweera). සියලු හිමිකම් ඇවිරිණි.

13 ශ්‍රේණිය A/L ICT · පාඩම 9

9.9 ශ්‍රිත: ගොඩනැගූ සහ පරිශීලක නිර්වචිත

විෂය නිර්දේශය 9.9. උප වැඩසටහනක් (ශ්‍රිතයක්) එක් කාර්යයක් කරන නම් කළ කොටසකි. Python හි **ගොඩනැගූ** ශ්‍රිත (len, max, round) ඇති අතර **පරාමිති**, **ප්‍රතිදාන අගය** සහ **පෙරනිමි අගය** සහිත **පරිශීලක නිර්වචිත** ශ්‍රිත ලිවිය හැක. විචල්‍යයක **විෂය පථය** එය භාවිත කළ හැකි තැන තීරණය කරයි: **local** විචල්‍ය ශ්‍රිතය තුළ පමණි; **global** විචල්‍ය සම්පූර්ණ වැඩසටහන පුරා ජීවත් වේ.

1. ශ්‍රිතයක ව්‍යුහය



ශ්‍රිතයක කොටස් සහ විෂය පථය

- def name(parameters): ශ්‍රිතය ආරම්භ කරයි; indent කළ ශරීරය පසුව.
- **පරාමිති** අගයන් ලබා ගනී; කැඳවීමේදී යවන අගයන් **arguments** ය.
- **return** අගයක් කැඳවන්නාට යවයි; නැත්නම් None.
- **පෙරනිමි අගය** (price=100) argument එක නොදුන් විට භාවිත වේ.

🐍 පරාමිති, ප්‍රතිදාන අගය, පෙරනිමි සහ විෂය පථය

```
count = 0                          # global variable

def total_price(qty, price=100):   # price has a default value
    return qty * price             # return value

def add_visit():
    global count
    count += 1

def show_local():
    count = 99                     # local: hides the global
    return count

print(total_price(3))
print(total_price(3, 250))
add_visit(); add_visit()
print("Local:", show_local(), "Global:", count)
print(len("Lanka"), max(4, 9, 2), round(3.567, 1))   # built-in
```

▶ **ප්‍රතිදානය**

```
300
750
Local: 99 Global: 2
5 9 3.6
```

🧱 **විසඳූ උදාහරණය: functions වැඩසටහනේ මුද්‍රිත පේළි පැහැදිලි කරන්න.**
total_price(3) පෙරනිමි 100 → **300**; total_price(3, 250) → **750**. add_visit() දෙවරක් **global** count 2 කරයි. show_local() හි **local** count = 99 ශ්‍රිතය අවසන් වූ විට නැති වේ; “Local: 99 Global: 2”. len = 5, max = 9, round = 3.6.

Local විචල්‍යය	Global විචල්‍යය
ශ්‍රිතය තුළ සාදයි	සියලු ශ්‍රිතවලින් පිටත
ශ්‍රිතය ධාවනය වන තුරු පමණි	වැඩසටහන අවසන් වන තුරු
එම ශ්‍රිතයේ පමණි	සෑම තැනම කියවිය හැක; global මගින් වෙනස් කළ හැක

💡 global විචල්‍යවලට වඩා පරාමිති සහ return භාවිත කරන්න.

🚀 **විෂය නිර්දේශයෙන් ඔබ්බට (2026 යාවත්කාලීනය)** *විභාගයට අවශ්‍ය නැත – දැනුම සඳහා කියවන්න.* def total_price(qty: int, price: float = 100) -> float වැනි type hints 2026 කේතයේ සුලබයි.

ප්‍රධාන පද

උප වැඩසටහන – කාර්යයක් කරන නම් කළ කේත කොටස.

ගොඩනැගූ ශ්‍රිතය – len, max වැනි Python සපයන ශ්‍රිතය.

පරිශීලක නිර්වචිත ශ්‍රිතය – def මගින් ක්‍රමලේඛකයා ලියන ශ්‍රිතය.

පරාමිතිය – අගයක් ලබා ගන්නා ශ්‍රිත නිර්වචනයේ විචල්‍යය.

Argument – කැඳවීමේදී ශ්‍රිතයට යවන සැබෑ අගය.

ප්‍රතිදාන අගය – return මගින් කැඳවන්නාට යවන අගය.

පෙරනිමි අගය – argument නොදුන් විට පරාමිතිය ගන්නා අගය.

විෂය පථය – විචල්‍යයක් භාවිත කළ හැකි වැඩසටහනේ කොටස.

Local විචල්‍යය – ශ්‍රිතය තුළ පමණක් පවතින විචල්‍යය.

Global විචල්‍යය – සම්පූර්ණ වැඩසටහනට ශ්‍රිතවලින් පිටත නිර්වචිත විචල්‍යය.

ආයු කාලය – විචල්‍යයක් මතකයේ පවතින කාලය.

ශ්‍රිත කැඳවීම – නමින් ශ්‍රිතයක් ධාවනය කරන ප්‍රකාශය.

ආදර්ශ ප්‍රශ්න සහ පිළිතුරු

1. ගොඩනැගූ සහ පරිශීලක?

> **පිළිතුර:** Python සමඟ; def මගින්.

2. පරාමිතිය සහ argument?

> **පිළිතුර:** නිර්වචනයේ; යවන අගය.

3. total_price(3)?

> **පිළිතුර:** 300.

4. Global 2 ඇයි?

> **පිළිතුර:** local එක පමණක් වෙනස් වේ.

5. local ආයු කාලය?

> **පිළිතුර:** ශ්‍රිතය ධාවනය වන තුරු.

6. return ඇයි?

> **පිළිතුර:** ප්‍රතිඵල ආපසු යැවීමට.

බහුවරණ ප්‍රශ්න

1. len, max

A. modules only
B. keywords
C. built-in functions
D. user-defined functions
E. constants

2. 250

A. argument
B. global
C. return value
D. parameter name
E. default value

3. total_price(3)

A. 100
B. None
C. 300
D. 3
E. 750

4. total_price(3, 250)

A. None
B. 750
C. 253
D. 300
E. 250

5. count

A. 99
B. 0
C. 1
D. 3
E. 2

6. show_local

A. 2
B. None
C. 99
D. 1
E. 0

7. ශ්‍රිතය තුළ

A. global
B. constant
C. built-in
D. local
E. default

8. round

A. 3.57
B. 3.0
C. 4
D. 3.5
E. 3.6

9. “කාර්යයක් කරන නම් කළ කේත කොටස” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ආයු කාලය
B. විෂය පථය
C. ප්‍රතිදාන අගය
D. උප වැඩසටහන
E. පරිශීලක නිර්වචිත ශ්‍රිතය

10. “len, max වැනි Python සපයන ශ්‍රිතය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. පෙරනිමි අගය
B. ගොඩනැගූ ශ්‍රිතය
C. Local විචල්‍යය
D. පරාමිතිය
E. ශ්‍රිත කැඳවීම

11. “def මගින් ක්‍රමලේඛකයා ලියන ශ්‍රිතය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. පරිශීලක නිර්වචිත ශ්‍රිතය
B. Global විචල්‍යය
C. Argument
D. විෂය පථය
E. උප වැඩසටහන

12. “අගයක් ලබා ගන්නා ශ්‍රිත නිර්වචනයේ විචල්‍යය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ආයු කාලය
B. ප්‍රතිදාන අගය
C. ගොඩනැගූ ශ්‍රිතය
D. පරාමිතිය
E. Local විචල්‍යය

13. “කැඳවීමේදී ශ්‍රිතයට යවන සැබෑ අගය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Global විචල්‍යය
B. පරිශීලක නිර්වචිත ශ්‍රිතය
C. Argument
D. ශ්‍රිත කැඳවීම
E. පෙරනිමි අගය

14. “return මගින් කැඳවන්නාට යවන අගය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. පරාමිතිය
B. ප්‍රතිදාන අගය
C. උප වැඩසටහන
D. විෂය පථය
E. ආයු කාලය

15. “argument නොදුන් විට පරාමිතිය ගන්නා අගය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. පෙරනිමි අගය
B. ශ්‍රිත කැඳවීම
C. Local විචල්‍යය
D. Argument
E. ගොඩනැගූ ශ්‍රිතය

16. “විචල්‍යයක් භාවිත කළ හැකි වැඩසටහනේ කොටස” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. විෂය පථය
B. Global විචල්‍යය
C. උප වැඩසටහන
D. ප්‍රතිදාන අගය
E. පරිශීලක නිර්වචිත ශ්‍රිතය

17. “ශ්‍රිතය තුළ පමණක් පවතින විචල්‍යය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. පරාමිතිය
B. Local විචල්‍යය
C. ආයු කාලය
D. ගොඩනැගූ ශ්‍රිතය
E. පෙරනිමි අගය

18. “සම්පූර්ණ වැඩසටහනට ශ්‍රිතවලින් පිටත නිර්වචිත විචල්‍යය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. පරිශීලක නිර්වචිත ශ්‍රිතය
B. ශ්‍රිත කැඳවීම
C. විෂය පථය
D. Global විචල්‍යය
E. Argument

19. “විචල්‍යයක් මතකයේ පවතින කාලය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Local විචල්‍යය
B. උප වැඩසටහන
C. ප්‍රතිදාන අගය
D. පරාමිතිය
E. ආයු කාලය

20. “නමින් ශ්‍රිතයක් ධාවනය කරන ප්‍රකාශය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. පෙරනිමි අගය
B. Argument
C. ගොඩනැගූ ශ්‍රිතය
D. Global විචල්‍යය
E. ශ්‍රිත කැඳවීම

ව්‍යුහගත ප්‍රශ්න

1. ශ්‍රිත වර්ග දෙක. *(ලකුණු 3)*

> len; def.

2. කොටස් නම් කරන්න. *(ලකුණු 5)*

> def; නම; පරාමිති; පෙරනිමි; return.

3. local සහ global. *(ලකුණු 3)*

> තැන; ආයු කාලය; භාවිතය.

4. area() ශ්‍රිතය. *(ලකුණු 3)*

> def; return; කැඳවීම්.

5. show_local(). *(ලකුණු 3)*

> local; වෙනස් නොවේ; global.

සාරාංශය

- ශ්‍රිත නැවත භාවිත කළ හැකි කොටස් ය.
- පරාමිති arguments ලබා ගනී.
- return අගයක් ආපසු යවයි.
- පෙරනිමි අගයන් නැති arguments පුරවයි.
- Local විචල්‍ය ශ්‍රිතය තුළ පමණි.

MCQ පිළිතුරු – 9.9

1. **C** – ගොඩනැගූ.
2. **A** – Argument.
3. **C** – 3 × 100.
4. **B** – 3 × 250.
5. **E** – දෙවරක්.
6. **C** – local අගය.
7. **D** – Local.
8. **E** – දශම 1.
9. **D** – උප වැඩසටහන – කාර්යයක් කරන නම් කළ කේත කොටස.
10. **B** – ගොඩනැගූ ශ්‍රිතය – len, max වැනි Python සපයන ශ්‍රිතය.
11. **A** – පරිශීලක නිර්වචිත ශ්‍රිතය – def මගින් ක්‍රමලේඛකයා ලියන ශ්‍රිතය.
12. **D** – පරාමිතිය – අගයක් ලබා ගන්නා ශ්‍රිත නිර්වචනයේ විචල්‍යය.
13. **C** – Argument – කැඳවීමේදී ශ්‍රිතයට යවන සැබෑ අගය.
14. **B** – ප්‍රතිදාන අගය – return මගින් කැඳවන්නාට යවන අගය.
15. **A** – පෙරනිමි අගය – argument නොදුන් විට පරාමිතිය ගන්නා අගය.
16. **A** – විෂය පථය – විචල්‍යයක් භාවිත කළ හැකි වැඩසටහනේ කොටස.
17. **B** – Local විචල්‍යය – ශ්‍රිතය තුළ පමණක් පවතින විචල්‍යය.
18. **D** – Global විචල්‍යය – සම්පූර්ණ වැඩසටහනට ශ්‍රිතවලින් පිටත නිර්වචිත විචල්‍යය.
19. **E** – ආයු කාලය – විචල්‍යයක් මතකයේ පවතින කාලය.
20. **E** – ශ්‍රිත කැඳවීම – නමින් ශ්‍රිතයක් ධාවනය කරන ප්‍රකාශය.

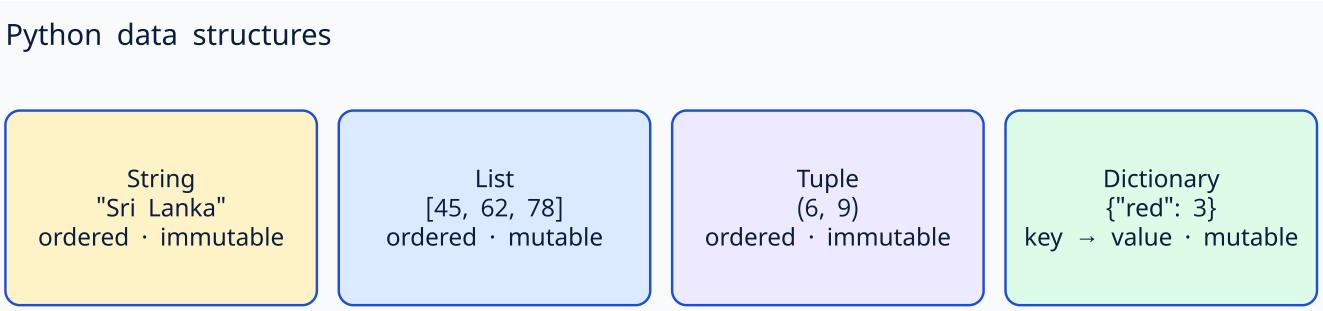
© 2026 schoolict.net (Ruwan Mangala Suraweera). සියලු හිමිකම් ඇවිරිණි.

13 ශ්‍රේණිය A/L ICT · පාඩම 9

9.10 දත්ත ව්‍යුහ: තන්තු, ලැයිස්තු, Tuples සහ ශබ්දකෝෂ

විෂය නිර්දේශය 9.10. දත්ත ව්‍යුහයක් එක් නමක් යටතේ අගයන් බොහොමයක් ගබඩා කරයි. Python හි **තන්තු** (පෙළ), **ලැයිස්තු** (වෙනස් කළ හැකි අනුක්‍රම), **tuples** (ස්ථිර අනුක්‍රම) සහ **ශබ්දකෝෂ** (key → value යුගල) ඇත. ඒවා සමඟ **indexing**, **slicing** සහ **methods** භාවිත කරමු.

1. ව්‍යුහ හතර



තන්තු, ලැයිස්තු, tuple සහ ශබ්දකෝෂ

ව්‍යුහය	වරහන්	පිළිවෙළ?	වෙනස් කළ හැකිද?
තන්තුව	" "	ඔව්	නැත
ලැයිස්තුව	[ ]	ඔව්	ඔව්
Tuple	( )	ඔව්	නැත
ශබ්දකෝෂය	{key: value}	ඇතුළත් කිරීම අනුව	ඔව්

🐍 තන්තු, ලැයිස්තු, tuples සහ ශබ්දකෝෂ

```
s = "Sri Lanka"
print(s[0], s[-1], s[4:9], len(s), s.upper())

marks = [45, 78, 62]
marks.append(90)
marks.sort()
print(marks, marks[1:3], sum(marks))

point = (6, 9)                 # tuple: cannot be changed
print(point[0] + point[1])

words = "red blue red green red".split()
freq = {}
for w in words:
    freq[w] = freq.get(w, 0) + 1
print(freq)
print(sorted(freq.items()))
```

▶ ප්‍රතිදානය

```
S a Lanka 9 SRI LANKA
[45, 62, 78, 90] [62, 78] 275
15
{'red': 3, 'blue': 1, 'green': 1}
[('blue', 1), ('green', 1), ('red', 3)]
```

🧱 **විසඳූ උදාහරණය: s[4:9], අනුපිළිවෙළ ලැයිස්තුව සහ වචන ගණන් පැහැදිලි කරන්න.** "Sri Lanka" අක්ෂර 9 (0–8); s[4:9] = **Lanka**. append, sort පසු **[45, 62, 78, 90]**; [1:3] = [62, 78]; එකතුව **275**. red **3**, blue 1, green 1.

- ලැයිස්තු methods: append, insert, remove, pop, sort, len().
- තන්තු methods: upper, lower, split, strip, find, replace.
- ශබ්දකෝෂ: d[key], d.get(), keys(), values(), items().

💡 සෘණ index අගින් ගණන් කරයි: s[-1] අවසාන අයිතමයයි.

🚀 **විෂය නිර්දේශයෙන් ඔබ්බට (2026 යාවත්කාලීනය)** *විභාගයට අවශ්‍ය නැත – දැනුම සඳහා කියවන්න.* 2026 web API භාවිත කරන JSON සෘජුවම Python ශබ්දකෝෂ සහ ලැයිස්තුවලට ගැළපේ.

ප්‍රධාන පද

දත්ත ව්‍යුහය – අගයන් බොහොමයක් එක් නමක් යටතේ ගබඩා කිරීම.

තන්තුව – අක්ෂරවල පිළිවෙළ වෙනස් කළ නොහැකි අනුක්‍රමය.

ලැයිස්තුව – හතරැස් වරහන් තුළ ලියන වෙනස් කළ හැකි අනුක්‍රමය.

Tuple – රවුම් වරහන් තුළ වෙනස් කළ නොහැකි අනුක්‍රමය.

ශබ්දකෝෂය – key සහ value යුගල එකතුව.	**Index** – 0 සිට ආරම්භ වන අයිතමයේ ස්ථාන අංකය.
Slice – start:stop මගින් ගන්නා අනුක්‍රම කොටස.	**වෙනස් කළ හැකි** – සෑදූ පසු වෙනස් කළ හැකි.
වෙනස් කළ නොහැකි – සෑදූ පසු වෙනස් කළ නොහැකි.	**Method** – append වැනි වස්තුවකට අයත් ශ්‍රිතය.
Key – ශබ්දකෝෂයක අගයක් සොයන අනන්‍ය ලේබලය.	**සම්බන්ධ කිරීම** – + මගින් තන්තු හෝ ලැයිස්තු එකතු කිරීම.

ආදර්ශ ප්‍රශ්න සහ පිළිතුරු

1. ලැයිස්තුව සහ tuple?

> **පිළිතුර:** ලැයිස්තු වෙනස් වේ; tuple නොවේ.

2. s[0], s[-1]?

> **පිළිතුර:** S සහ a.

3. append, sort පසු?

> **පිළිතුර:** [45, 62, 78, 90].

4. red වාර?

> **පිළිතුර:** 3.

5. freq.get(w, 0)?

> **පිළිතුර:** ගණන හෝ 0.

6. point සඳහා tuple ඇයි?

> **පිළිතුර:** ඛණ්ඩාංක වෙනස් නොවිය යුතුය.

බහුවරණ ප්‍රශ්න

1. s[4:9]

A. Sri
B. anka
C. Lank
D. Sri L
E. Lanka

2. len

A. 8
B. 2
C. 9
D. 7
E. 10

3. [1:3]

A. [45, 62, 78]
B. [62, 78]
C. [62, 78, 90]
D. [78, 90]
E. [45, 62]

4. sum

A. 230
B. 90
C. 275
D. 285
E. 265

5. වෙනස් නොවන

A. tuple
B. list
C. none
D. dictionary
E. set of lists

6. red

A. 2
B. 5
C. 3
D. 0
E. 1

7. key → value

A. file
B. dictionary
C. tuple
D. string
E. list only

8. (6, 9)

A. 3
B. 6
C. 54
D. 69
E. 15

9. “අගයන් බොහොමයක් එක් නමක් යටතේ ගබඩා කිරීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. දත්ත ව්‍යුහය
B. වෙනස් කළ හැකි
C. Index
D. ලැයිස්තුව
E. Key

10. “අක්ෂරවල පිළිවෙළ වෙනස් කළ නොහැකි අනුක්‍රමය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. තන්තුව
B. Slice
C. Tuple
D. වෙනස් කළ නොහැකි
E. සම්බන්ධ කිරීම

11. “හතරැස් වරහන් තුළ ලියන වෙනස් කළ හැකි අනුක්‍රමය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Method
B. වෙනස් කළ හැකි
C. දත්ත ව්‍යුහය
D. ලැයිස්තුව
E. ශබ්දකෝෂය

12. “රවුම් වරහන් තුළ වෙනස් කළ නොහැකි අනුක්‍රමය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Index
B. Key
C. Tuple
D. තන්තුව
E. වෙනස් කළ නොහැකි

13. “key සහ value යුගල එකතුව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Method
B. ශබ්දකෝෂය
C. සම්බන්ධ කිරීම
D. Slice
E. ලැයිස්තුව

14. “0 සිට ආරම්භ වන අයිතමයේ ස්ථාන අංකය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. වෙනස් කළ හැකි
B. Index
C. Key
D. Tuple
E. දත්ත ව්‍යුහය

15. “start:stop මගින් ගන්නා අනුක්‍රම කොටස” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ශබ්දකෝෂය
B. තන්තුව
C. සම්බන්ධ කිරීම
D. Slice
E. වෙනස් කළ නොහැකි

16. “සෑදූ පසු වෙනස් කළ හැකි” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. වෙනස් කළ හැකි
B. දත්ත ව්‍යුහය
C. ලැයිස්තුව
D. Method
E. Index

17. “සෑදූ පසු වෙනස් කළ නොහැකි” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Key
B. Tuple
C. තන්තුව
D. Slice
E. වෙනස් කළ නොහැකි

18. “append වැනි වස්තුවකට අයත් ශ්‍රිතය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ලැයිස්තුව
B. වෙනස් කළ හැකි
C. සම්බන්ධ කිරීම
D. Method
E. ශබ්දකෝෂය

19. “ශබ්දකෝෂයක අගයක් සොයන අනන්‍ය ලේඛලය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Tuple
B. Index
C. දත්ත ව්‍යුහය
D. වෙනස් කළ නොහැකි
E. Key

20. “+ මගින් තන්තු හෝ ලැයිස්තු එකතු කිරීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Method
B. Slice
C. ශබ්දකෝෂය
D. සම්බන්ධ කිරීම
E. තන්තුව

ව්‍යුහගත ප්‍රශ්න

1. ව්‍යුහ හතර සසඳන්න. *(ලකුණු 4)*

> වරහන්; පිළිවෙළ; වෙනස්; key.

2. ප්‍රතිදානය. *(ලකුණු 4)*

> S; a; Lanka; SRI LANKA.

3. ලැයිස්තු ක්‍රේතය. *(ලකුණු 3)*

> append; sort; sum.

4. වචන ගණන් කිරීම. *(ලකුණු 4)*

> split; for; get.

5. mutable සහ immutable. *(ලකුණු 2)*

> ලැයිස්තුව; tuple.

සාරාංශය

- තන්තු පෙළ රඳවයි; වෙනස් නොවේ.
- ලැයිස්තු පිළිවෙළ සහ වෙනස් කළ හැක.
- Tuples පිළිවෙළ සහ ස්ථිරයි.
- ශබ්දකෝෂ keys අගයන්ට සම්බන්ධ කරයි.
- Index 0 සිට; slice අවසානය ඇතුළත් නොවේ.

MCQ පිළිතුරු – 9.10

1. **E** – 4 සිට 8.
2. **C** – හිස්තැනද ගණන් වේ.
3. **B** – 1 සහ 2.
4. **C** – එකතුව.
5. **A** – Tuple.
6. **C** – තුන් වරක්.
7. **B** – ශබ්දකෝෂය.
8. **E** – 6 + 9.
9. **A** – දත්ත ව්‍යුහය – අගයන් බොහොමයක් එක් නමක් යටතේ ගබඩා කිරීම.
10. **A** – තන්තුව – අක්ෂරවල පිළිවෙළ වෙනස් කළ නොහැකි අනුක්‍රමය.
11. **D** – ලැයිස්තුව – හතරැස් වරහන් තුළ ලියන වෙනස් කළ හැකි අනුක්‍රමය.
12. **C** – Tuple – රවුම් වරහන් තුළ වෙනස් කළ නොහැකි අනුක්‍රමය.
13. **B** – ශබ්දකෝෂය – key සහ value යුගල එකතුව.
14. **B** – Index – 0 සිට ආරම්භ වන අයිතමයේ ස්ථාන අංකය.
15. **D** – Slice – start:stop මගින් ගන්නා අනුක්‍රම කොටස.
16. **A** – වෙනස් කළ හැකි – සෑදූ පසු වෙනස් කළ හැකි.
17. **E** – වෙනස් කළ නොහැකි – සෑදූ පසු වෙනස් කළ නොහැකි.
18. **D** – Method – append වැනි වස්තුවකට අයත් ශ්‍රිතය.
19. **E** – Key – ශබ්දකෝෂයක අගයක් සොයන අනන්‍ය ලේබලය.
20. **D** – සම්බන්ධ කිරීම – + මගින් තන්තු හෝ ලැයිස්තු එකතු කිරීම.

© 2026 schoolict.net (Ruwan Mangala Suraweera). සියලු හිමිකම් ඇවිරිණි.

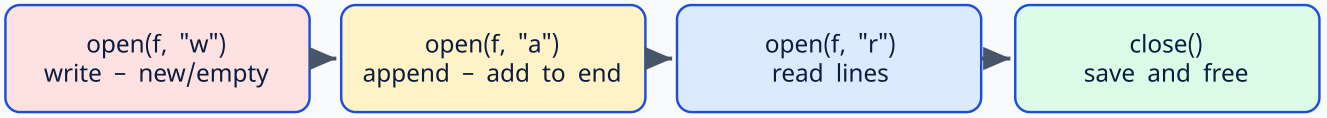
13 ශ්‍රේණිය A/L ICT · පාඩම 9

9.11 ගොනු හැසිරවීම: විවෘත, කියවීම, ලිවීම, එකතු කිරීම සහ වැසීම

විෂය නිර්දේශය 9.11. වැඩසටහන අවසන් වූ විට විචල්‍ය නැති වේ; **ගොනු** දත්ත ගබඩාවේ තබා ගනී. Python හි **mode** එකක් සමඟ ගොනුව **විවෘත** කර, **කියවා** හෝ **ලියා**, **වසන්න**. "w" ගොනුව සාදයි හෝ හිස් කරයි, "a" අගට **එකතු** කරයි, "r" කියවයි.

1. ගොනු modes

File modes



Opening in "w" mode erases an existing file – use "a" to keep old data.

r, w සහ a modes

Mode	අර්ථය	ගොනුව නැත්නම්
"r"	මුල සිට කියවීම	දෝෂය
"w"	ලිවීම – පැරණි අන්තර්ගතය මැකේ	සාදයි
"a"	එකතු කිරීම – අගට	සාදයි

🐍 පෙළ ගොනුවක් ලිවීම, එක් කිරීම සහ කියවීම

```
f = open("marks.txt", "w")      # write: creates or empties
f.write("Nimal,78\n")
f.write("Kamala,45\n")
f.close()

f = open("marks.txt", "a")      # append: adds to the end
f.write("Sunil,88\n")
f.close()

f = open("marks.txt", "r")      # read
total = 0
for line in f:
    name, mark = line.strip().split(",")
    total += int(mark)
    print(name, mark)
f.close()
print("Total:", total)
```

▶ ප්‍රතිදානය

```
Nimal 78
Kamala 45
Sunil 88
Total: 211
```

🧱 **විසඳූ උදාහරණය: ගොනුවේ පේළි තුනක් සහ Total 211 ඇයි?**
"w" Nimal සහ Kamala ලියයි; "a" Sunil එකතු කරයි. strip(), split(","), int(): 78 + 45 + 88 = **211**.

- read() මුළු ගොනුව; readline() එක් පේළියක්; readlines() ලැයිස්තුවක්.
- write() තන්තුවක් ලියයි – නව පේළියට "\n" එක් කරන්න.
- close() buffer එක සුරකියි, ගොනුව නිදහස් කරයි.

💡 with open(...) as f: දෝෂයක් වුවද ගොනුව ස්වයංක්‍රීයව වසයි.

🚀 විෂය නිර්දේශයෙන් ඔබ්බට (2026 යාවත්කාලීනය) *විභාගයට අවශ්‍ය නැත – දැනුම සඳහා කියවන්න.* 2026 දී බොහෝ යෙදුම් CSV හෝ JSON ගොනු භාවිත කරයි; csv සහ json modules ඒවා පහසුවෙන් කියවයි.

ප්‍රධාන පද

ගොනුව – ගබඩාවේ තබන නම් කළ දත්ත එකතුව.	**open()** – mode එකකින් ගොනුවක් විවෘත කරන ශ්‍රිතය.
close() – ගොනුව භාවිතය අවසන් කර සුරකින method.	**කියවීමේ mode** – පවතින ගොනුවක් කියවන r mode.
ලිවීමේ mode – ගොනුවක් සාදන හෝ හිස් කරන w mode.	**එකතු කිරීමේ mode** – ගොනුවේ අගට දත්ත එක් කරන a mode.
read() – මුළු ගොනුව තන්තුවක් ලෙස දෙන method.	**readline()** – ගොනුවේ එක් පේළියක් දෙන method.
write() – ගොනුවට තන්තුවක් ලියන method.	**නව පේළිය** – පෙළ පේළියක් අවසන් කරන \n අක්ෂරය.
පෙළ ගොනුව – .txt, .csv වැනි කියවිය හැකි අක්ෂර ගොනුව.	**ස්ථිර ගබඩාව** – වැඩසටහන අවසන් වූ පසුද දත්ත තබා ගන්නා ගබඩාව.

ආදර්ශ ප්‍රශ්න සහ පිළිතුරු

1. ගොනු ඇයි?

> **පිළිතුර:** දත්ත ස්ථිරව තබා ගැනීමට.

2. "w" සහ "a"?

> **පිළිතුර:** "w" මකයි; "a" එකතු කරයි.

3. "r" නැති ගොනුව?

> **පිළිතුර:** දෝෂයක්.

4. strip() ඇයි?

> **පිළිතුර:** නව පේළිය ඉවත් කිරීමට.

5. Total?

> **පිළිතුර:** 211.

6. වැසීම ඇයි?

> **පිළිතුර:** දත්ත සුරැකීමට.

බහුවරණ ප්‍රශ්න

1. අගට

A. "a"
B. "x" only
C. "w"
D. "rb"
E. "r"

2. "w"

A. gives an error
B. locks it forever
C. adds to the end
D. erases its old content
E. only reads it

3. Total

A. 88
B. 211
C. 123
D. 200
E. 166

4. පේළි

A. 4
B. 1
C. 3
D. 0
E. 2

5. split

A. ("Nimal", 78)
B. ["Nimal", "78"]
C. "Nimal78"
D. ["Nimal,78"]
E. [78]

6. එක් පේළියක්

A. open()
B. close()
C. read()
D. write()
E. readline()

7. "r"

A. creates it
B. raises an error
C. returns 0
D. appends
E. empties it

8. with

A. closes the file automatically
B. sorts lines
C. opens two files
D. deletes the file
E. reads only binary

9. “ගබඩාවේ තබන නම් කළ දත්ත එකතුව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. පෙළ ගොනුව
B. close()
C. ගොනුව
D. readline()
E. එකතු කිරීමේ mode

10. “mode එකකින් ගොනුවක් විවෘත කරන ශ්‍රිතය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. open()
B. read()
C. කියවීමේ mode
D. ස්ථිර ගබඩාව
E. write()

11. “ගොනුව භාවිතය අවසන් කර සුරකින method” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ගොනුව
B. ලිවීමේ mode
C. නව පේළිය
D. readline()
E. close()

12. “පවතින ගොනුවක් කියවන r mode” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. එකතු කිරීමේ mode
B. open()
C. write()
D. කියවීමේ mode
E. පෙළ ගොනුව

13. “ගොනුවක් සාදන හෝ හිස් කරන w mode” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. නව පේළිය
B. close()
C. read()
D. ස්ථිර ගබඩාව
E. ලිවීමේ mode

14. “ගොනුවේ අගට දත්ත එක් කරන a mode” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. readline()
B. ගොනුව
C. කියවීමේ mode
D. එකතු කිරීමේ mode
E. පෙළ ගොනුව

15. “මුළු ගොනුව තන්තුවක් ලෙස දෙන method” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ස්ථිර ගබඩාව
B. write()
C. open()
D. read()
E. ලිවීමේ mode

16. “ගොනුවේ එක් පේළියක් දෙන method” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. readline()
B. නව පේළිය
C. close()
D. එකතු කිරීමේ mode
E. ගොනුව

17. “ගොනුවට තන්තුවක් ලියන method” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. කියවීමේ mode
B. write()
C. පෙළ ගොනුව
D. read()
E. open()

18. “පෙළ පේළියක් අවසන් කරන \n අක්ෂරය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ස්ථිර ගබඩාව
B. close()
C. ලිවීමේ mode
D. readline()
E. නව පේළිය

19. “.txt, .csv වැනි කියවිය හැකි අක්ෂර ගොනුව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ගොනුව
B. write()
C. පෙළ ගොනුව
D. කියවීමේ mode
E. එකතු කිරීමේ mode

20. “වැඩසටහන අවසන් වූ පසුද දත්ත තබා ගන්නා ගබඩාව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ලිවීමේ mode
B. read()
C. ස්ථිර ගබඩාව
D. නව පේළිය
E. open()

ව්‍යුහගත ප්‍රශ්න

1. modes තුන. *(ලකුණු 3)*

> r; w; a.

2. නම් තුනක් ලියන්න. *(ලකුණු 3)*

> open; write; close.

3. total කියවන්න. *(ලකුණු 6)*

> open; loop; int.

4. 211 ඇයි? *(ලකුණු 3)*

> w; a; එකතුව.

5. වැසීම හේතු 2. *(ලකුණු 2)*

> සුරැකීම; නිදහස් කිරීම.

සාරාංශය

- ගොනු දත්ත ස්ථිරව තබයි.
- r, w, a සමඟ open().
- "w" මකයි; "a" එකතු කරයි.
- read(), readline() හෝ for.
- close() හෝ with.

MCQ පිළිතුරු – 9.11

1. **A** – Append.
2. **D** – හිස් කරයි.
3. **B** – 78 + 45 + 88.
4. **C** – 2 + 1.
5. **B** – තන්තු දෙකක්.
6. **E** – readline.
7. **B** – දෝෂය.
8. **A** – ස්වයංක්‍රීයව වසයි.
9. **C** – ගොනුව – ගබඩාවේ තබන නම් කළ දත්ත එකතුව.
10. **A** – open() – mode එකකින් ගොනුවක් විවෘත කරන ශ්‍රිතය.
11. **E** – close() – ගොනුව භාවිතය අවසන් කර සුරකින method.
12. **D** – කියවීමේ mode – පවතින ගොනුවක් කියවන r mode.
13. **E** – ලිවීමේ mode – ගොනුවක් සාදන හෝ හිස් කරන w mode.
14. **D** – එකතු කිරීමේ mode – ගොනුවේ අගට දත්ත එක් කරන a mode.
15. **D** – read() – මුළු ගොනුව තන්තුවක් ලෙස දෙන method.

16. **A** – readline() – ගොනුවේ එක් පේළියක් දෙන method.
17. **B** – write() – ගොනුවට තන්තුවක් ලියන method.
18. **E** – නව පේළිය – පෙළ පේළියක් අවසන් කරන \n අක්ෂරය.
19. **C** – පෙළ ගොනුව – .txt, .csv වැනි කියවිය හැකි අක්ෂර ගොනුව.
20. **C** – ස්ථිර ගබඩාව – වැඩසටහන අවසන් වූ පසුද දත්ත තබා ගන්නා ගබඩාව.

© 2026 schoolict.net (Ruwan Mangala Suraweera). සියලු හිමිකම් ඇවිරිණි.

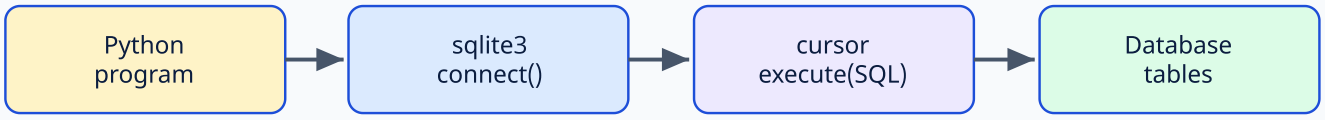
13 ශ්‍රේණිය A/L ICT · පාඩම 9

9.12 Python සහ දත්ත සමුදා: සම්බන්ධ, එකතු, වෙනස්, මැකීම, ලබා ගැනීම

විෂය නිර්දේශය 9.12. වැඩසටහනකට දත්ත සමුදායකට **සම්බන්ධ** වී SQL මගින් වාර්තා **එකතු**, **වෙනස්**, **මැකීම** සහ **ලබා ගැනීම** කළ හැක. Python හි **sqlite3** module එකට සේවාදායකයක් අවශ්‍ය නැත: connect() විවෘත කරයි, **cursor** execute() මගින් SQL ධාවනය කරයි, commit() සුරකියි, close() විසන්ධි කරයි.

1. පියවර

Python program and a database



INSERT, UPDATE, DELETE need commit(); SELECT rows come back as tuples.

දත්ත සමුදාය සමඟ Python වැඩසටහන

- **සම්බන්ධ**: con = sqlite3.connect("school.db").
- **Cursor**: cur = con.cursor() SQL ධාවනය කරයි.
- **එකතු / වෙනස් / මැකීම**: INSERT, UPDATE, DELETE පසු con.commit().
- **ලබා ගැනීම**: SELECT, පේළි හරහා loop; පසු con.close().

🐍 SQLite දත්ත සමුදාය සමඟ Python

```
import sqlite3
con = sqlite3.connect(":memory:")       # connect
cur = con.cursor()
cur.execute("CREATE TABLE Student (ID TEXT PRIMARY KEY, Name TEXT, Mark INTEGER)")
rows = [("S001", "Nimal", 78), ("S002", "Kamala", 45), ("S003", "Sunil", 88)]
cur.executemany("INSERT INTO Student VALUES (?, ?, ?)", rows)       # add
cur.execute("UPDATE Student SET Mark = 50 WHERE ID = ?", ("S002",))  # modify
cur.execute("DELETE FROM Student WHERE ID = ?", ("S003",))           # delete
con.commit()
for row in cur.execute("SELECT Name, Mark FROM Student ORDER BY Name"):  # retrieve
    print(row)
con.close()
```

▶ ප්‍රතිදානය

```
('Kamala', 50)
('Nimal', 78)
```

🧱 **විසඳූ උදාහරණය: Kamala 50 සහ Nimal 78 පමණක් ඇයි?**
පේළි තුනක් ඇතුළු වේ. UPDATE S002 ලකුණ 45 සිට **50**; DELETE S003 ඉවත් කරයි. ORDER BY Name ඉතිරි පේළි දෙක දෙයි.

💡 SQL වෙත අගයන් දැමීමට ? placeholders භාවිත කරන්න – SQL injection වළක්වයි.

🚀 **විෂය නිර්දේශයෙන් ඔබ්බට (2026 යාවත්කාලීනය)** *විභාගයට අවශ්‍ය නැත – දැනුම සඳහා කියවන්න.* 2026 web යෙදුම් SQLAlchemy, Django වැනි ORM භාවිත කරයි; යටින් connect–execute–commit එකමයි.

ප්‍රධාන පද

දත්ත සමුදා සම්බන්ධතාව – වැඩසටහනක් සහ දත්ත සමුදායක් අතර සබැඳිය.	**sqlite3** – SQLite සඳහා Python module.
connect() – සම්බන්ධතාවක් විවෘත කරන ශ්‍රිතය.	**Cursor** – SQL ධාවනය කර ප්‍රතිඵල රඳවන වස්තුව.
execute() – එක් SQL ප්‍රකාශයක් ධාවනය කරන method.	**commit()** – වෙනස්කම් ස්ථිරව සුරකින method.
INSERT – වාර්තාවක් එක් කරන SQL විධානය.	**UPDATE** – වාර්තා වෙනස් කරන SQL විධානය.
DELETE – වාර්තා ඉවත් කරන SQL විධානය.	**SELECT** – වාර්තා ලබා ගන්නා SQL විධානය.
Placeholder – SQL හි අගයක් ආරක්ෂිතව ගන්නා ? ලකුණ.	**SQL injection** – ආදානය හරහා හානිකර SQL ඇතුළු කරන ප්‍රහාරය.

ආදර්ශ ප්‍රශ්න සහ පිළිතුරු

1. මෙහෙයුම් හතර?

පිළිතුර: එකතු, වෙනස්, මැකීම, ලබා ගැනීම.

2. cursor?

පිළිතුර: SQL ධාවනය කරයි.

3. commit() ඇයි?

පිළිතුර: සුරැකීමට.

4. Kamala ලකුණ?

පිළිතුර: 50.

5. පේළි කීයද?

> **පිළිතුර:** 2.

6. ? ඇයි?

> **පිළිතුර:** SQL injection වැළැක්වීම.

බහුවරණ ප්‍රශ්න

1. module

A. sqlite3
B. mysql
C. os
D. csv
E. pandas

2. SQL ධාවනය

A. connection string
B. cursor
C. print
D. table
E. commit

3. සුරැකීම

A. commit()
B. cursor()
C. execute()
D. fetch()
E. close() only

4. Kamala

A. 88
B. 45
C. 0
D. 50
E. 78

5. DELETE

A. S003 Sunil
B. none
C. S002 Kamala
D. all records
E. S001 Nimal

6. පළමු පේළිය

A. ('S001', 'Nimal')
B. ('Sunil', 88)
C. ('Kamala', 45)
D. ('Nimal', 78)
E. ('Kamala', 50)

7. ලබා ගැනීම

A. UPDATE
B. INSERT
C. DELETE
D. SELECT
E. DROP

8. ?

A. overflow
B. syntax colouring
C. slow disks
D. viruses in files
E. SQL injection

9. “වැඩසටහනක් සහ දත්ත සමුදායක් අතර සබැඳිය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. UPDATE
B. commit()
C. දත්ත සමුදා සම්බන්ධතාව
D. Placeholder
E. connect()

10. “SQLite සඳහා Python module” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Cursor
B. sqlite3
C. SQL injection
D. INSERT
E. DELETE

11. “සම්බන්ධතාවක් විවෘත කරන ශ්‍රිතය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. SELECT
B. execute()
C. connect()
D. UPDATE
E. දත්ත සමුදා සම්බන්ධතාව

12. “SQL ධාවනය කර ප්‍රතිඵල රඳවන වස්තුව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. commit()
B. Placeholder
C. DELETE
D. sqlite3
E. Cursor

13. “එක් SQL ප්‍රකාශයක් ධාවනය කරන method” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. SELECT
B. execute()
C. INSERT
D. SQL injection
E. connect()

14. “වෙනස්කම් ස්ථිරව සුරකින method” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. දත්ත සමුදා සම්බන්ධතාව
B. commit()
C. UPDATE
D. Placeholder
E. Cursor

15. “වාර්තාවක් එක් කරන SQL විධානය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. sqlite3
B. execute()
C. DELETE
D. INSERT
E. SQL injection

16. “වාර්තා වෙනස් කරන SQL විධානය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. UPDATE
B. commit()
C. SELECT
D. දත්ත සමුදා සම්බන්ධතාව
E. connect()

17. “වාර්තා ඉවත් කරන SQL විධානය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. sqlite3
B. INSERT
C. DELETE
D. Cursor
E. Placeholder

18. “වාර්තා ලබා ගන්නා SQL විධානය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. SQL injection
B. UPDATE
C. connect()
D. execute()
E. SELECT

19. “SQL හි අගයක් ආරක්ෂිතව ගන්නා ? ලකුණ” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. commit()
B. දත්ත සමුදා සම්බන්ධතාව
C. Placeholder
D. Cursor
E. DELETE

20. “ආදානය හරහා හානිකර SQL ඇතුළු කරන ප්‍රහාරය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. SELECT
B. execute()
C. sqlite3
D. SQL injection
E. INSERT

ව්‍යුහගත ප්‍රශ්න

1. පියවර. *(ලකුණු 5)*

> connect; cursor; execute; commit; close.

2. INSERT සහ SELECT. *(ලකුණු 4)*

> INSERT; commit; SELECT.

3. පේළි දෙක. *(ලකුණු 4)*

> 3; DELETE; UPDATE.

4. commit සහ close. *(ලකුණු 2)*

> සුරැකීම; නිදහස් කිරීම.

5. SQL injection. *(ලකුණු 2)*

> හානිකර ආදානය; placeholders.

සාරාංශය

- connect() විවෘත කරයි.
- cursor SQL ධාවනය කරයි.
- INSERT, UPDATE, DELETE වෙනස් කරයි.
- SELECT පේළි ලබා දෙයි.
- commit() සුරකියි; ආරක්ෂාවට ?.

MCQ පිළිතුරු – 9.12

1. **A** – sqlite3.
2. **B** – Cursor.
3. **A** – Commit.
4. **D** – UPDATE.
5. **A** – S003.
6. **E** – ORDER BY Name.
7. **D** – SELECT.
8. **E** – Injection.
9. **C** – දත්ත සමුදා සම්බන්ධතාව – වැඩසටහනක් සහ දත්ත සමුදායක් අතර සබැඳිය.
10. **B** – sqlite3 – SQLite සඳහා Python module.
11. **C** – connect() – සම්බන්ධතාවක් විවෘත කරන ශ්‍රිතය.
12. **E** – Cursor – SQL ධාවනය කර ප්‍රතිඵල රඳවන වස්තුව.
13. **B** – execute() – එක් SQL ප්‍රකාශයක් ධාවනය කරන method.
14. **B** – commit() – වෙනස්කම් ස්ථිරව සුරකින method.
15. **D** – INSERT – වාර්තාවක් එක් කරන SQL විධානය.
16. **A** – UPDATE – වාර්තා වෙනස් කරන SQL විධානය.
17. **C** – DELETE – වාර්තා ඉවත් කරන SQL විධානය.
18. **E** – SELECT – වාර්තා ලබා ගන්නා SQL විධානය.
19. **C** – Placeholder – SQL හි අගයක් ආරක්ෂිතව ගන්නා ? ලකුණ.
20. **D** – SQL injection – ආදානය හරහා හානිකර SQL ඇතුළු කරන ප්‍රහාරය.

© 2026 schoolict.net (Ruwan Mangala Suraweera). සියලු හිමිකම් ඇවිරිණි.

13 ශ්‍රේණිය A/L ICT · පාඩම 9

9.13 සෙවීම සහ වර්ග කිරීම: අනුක්‍රමික සෙවීම සහ Bubble Sort

විෂය නිර්දේශය 9.13. අනුක්‍රමික (රේඛීය) සෙවීම ඉලක්කය හමු වන තුරු හෝ ලැයිස්තුව අවසන් වන තුරු මුල සිට එකින් එක පරීක්ෂා කරයි. **Bubble sort** යාබද අයිතම සසඳා වැරදි පිළිවෙළ නම් හුවමාරු කරයි; සෑම **pass** එකකටම ඉතිරි විශාලතම අගය අගට “bubble” වේ.

1. අනුක්‍රමික සෙවීම

🐍 අනුක්‍රමික සෙවීම

```
def linear_search(items, target):
    for i in range(len(items)):
        if items[i] == target:
            return i
    return -1

data = [34, 7, 23, 32, 5, 62]
print(linear_search(data, 23))
print(linear_search(data, 99))
```

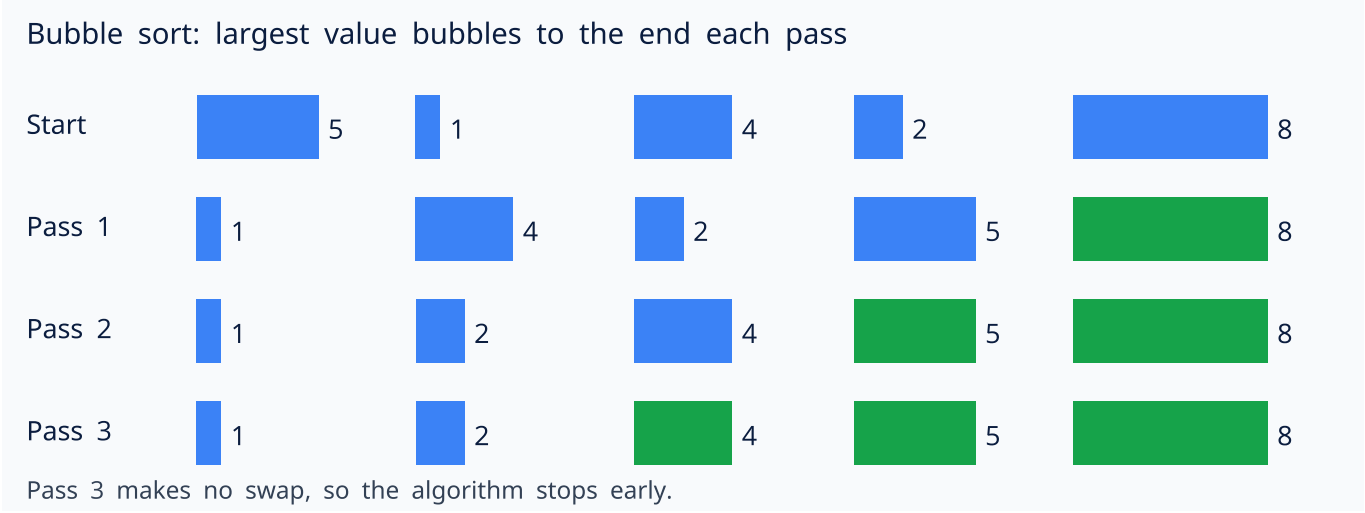
▶ ප්‍රතිදානය

```
2
-1
```

ඉලක්කය	ප්‍රතිඵල index	සැසඳීම්
23	2	3
99	−1	6

🧮 **විසඳූ උදාහරණය: [34, 7, 23, 32, 5, 62] හි 23 සහ 99 සෙවීමට සැසඳීම් කීයද?**
23: 34, 7, 23 – index **2**, සැසඳීම් **3**. 99: අයිතම **6** සමඟම සසඳා **−1** දෙයි.

2. Bubble sort



Bubble sort හි pass 1

🐍 සෑම pass එකක්ම පෙන්වන bubble sort

```
def bubble_sort(a):
    n = len(a)
    for p in range(n - 1):
        swapped = False
        for j in range(n - 1 - p):
            if a[j] > a[j + 1]:
                a[j], a[j + 1] = a[j + 1], a[j]
                swapped = True
        print("Pass", p + 1, a)
        if not swapped:
            break
    return a

bubble_sort([5, 1, 4, 2, 8])
```

▶ ප්‍රතිදානය

```
Pass 1 [1, 4, 2, 5, 8]
Pass 2 [1, 2, 4, 5, 8]
Pass 3 [1, 2, 4, 5, 8]
```

Pass	pass එකට පසු ලැයිස්තුව
1	[1, 4, 2, 5, 8]
2	[1, 2, 4, 5, 8]
3	[1, 2, 4, 5, 8]

🧱 විසඳූ උදාහරණය: [5, 1, 4, 2, 8] bubble sort කරන්න.
Pass 1: 5↔1, 5↔4, 5↔2 → [1, 4, 2, 5, 8]. Pass 2: 4↔2 → [1, 2, 4, 5, 8]. Pass 3 හුවමාරු නැත; නවතී. මුළු හුවමාරු **4**.

💡 අයිතම n සඳහා bubble sort උපරිම pass n − 1; සෙවීම උපරිම සැසඳීම් n.

🚀 විෂය නිර්දේශයෙන් ඔබ්බට (2026 යාවත්කාලීනය) *විභාගයට අවශ්‍ය නැත – දැනුම සඳහා කියවන්න.* Python sorted() Timsort භාවිත කරයි; bubble sort ඉගෙනීමට සම්භාව්‍යයි.

ප්‍රධාන පද

සෙවීම – අගයක් ලැයිස්තුවේ කොතැනද යන්න සොයා ගැනීම.

අනුක්‍රමික සෙවීම – මුල සිට එකින් එක පරීක්ෂා කිරීම.

ඉලක්කය – සොයන අගය.

සැසඳීම – අයිතමයක් ඉලක්කය හෝ අසල්වැසියා සමඟ පරීක්ෂාව.

වර්ග කිරීම – ආරෝහණ හෝ අවරෝහණ පිළිවෙළට සැකසීම.

Bubble sort – වැරදි පිළිවෙළේ යාබද අයිතම හුවමාරු කරන sort.

Pass – bubble sort හි ලැයිස්තුව හරහා එක් සම්පූර්ණ ගමන.

හුවමාරුව – අයිතම දෙකක ස්ථාන මාරු කිරීම.

Swapped flag – හුවමාරු නැති pass එකකදී නවත්වන Boolean.

ආරෝහණ පිළිවෙළ – කුඩාම සිට විශාලතම.

−1 ආපසු – ඉලක්කය හමු නොවූ බවට සංඥාව.

ඇල්ගොරිතම කාර්යක්ෂමතාව – දත්ත වැඩි වන විට වැඩ වැඩි වන ආකාරය.

ආදර්ශ ප්‍රශ්න සහ පිළිතුරු

1. අනුක්‍රමික සෙවීම?

පිළිතුර: එකින් එක පරීක්ෂාව.

2. 23 index?

පිළිතුර: 2.

3. 99 සැසඳීම්?

පිළිතුර: 6, −1.

4. pass 1 පසු?

පිළිතුර: [1, 4, 2, 5, 8].

5. pass 3 පසු ඇයි?

පිළිතුර: හුවමාරු නැත.

6. අයිතම 5 උපරිම pass?

> **පිළිතුර:** 4.

බහුවරණ ප්‍රශ්න

1. 23

A. 2
B. −1
C. 3
D. 23
E. 1

2. 23

A. 2
B. 1
C. 4
D. 3
E. 6

3. 99

A. 6
B. 99
C. −1
D. None
E. 0

4. pass 1

A. [5, 4, 2, 1, 8]
B. [1, 4, 2, 5, 8]
C. [8, 5, 4, 2, 1]
D. [1, 2, 4, 5, 8]
E. [1, 5, 4, 2, 8]

5. හුවමාරු

A. 4
B. 3
C. 5
D. 6
E. 2

6. සසඳයි

A. random pairs
B. every item with the target
C. the middle item
D. first and last only
E. neighbouring items

7. flag

A. swap faster
B. skip pass 1
C. search for a value
D. stop early when a pass has no swaps
E. sort descending

8. උපරිම

A. n + 1
B. n^2
C. n − 1
D. 2n
E. n

9. “අගයක් ලැයිස්තුවේ කොතැනද යන්න සොයා ගැනීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Bubble sort
B. −1 ආපසු
C. හුවමාරුව
D. ඉලක්කය
E. සෙවීම

10. “මුල සිට එකින් එක පරීක්ෂා කිරීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Pass
B. අනුක්‍රමික සෙවීම
C. Swapped flag
D. සැසඳීම
E. ඇල්ගොරිතම කාර්යක්ෂමතාව

11. “සොයන අගය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ඉලක්කය
B. සෙවීම
C. ආරෝහණ පිළිවෙළ
D. හුවමාරුව
E. වර්ග කිරීම

12. “අයිතමයක් ඉලක්කය හෝ අසල්වැසියා සමඟ පරීක්ෂාව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Swapped flag
B. −1 ආපසු
C. සැසඳීම
D. Bubble sort
E. අනුක්‍රමික සෙවීම

13. “ආරෝහණ හෝ අවරෝහණ පිළිවෙළට සැකසීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ආරෝහණ පිළිවෙළ
B. වර්ග කිරීම
C. ඉලක්කය
D. ඇල්ගොරිතම කාර්යක්ෂමතාව
E. Pass

14. “වැරදි පිළිවෙළේ යාබද අයිතම හුවමාරු කරන sort” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. −1 ආපසු
B. Bubble sort
C. සැසඳීම
D. හුවමාරුව
E. සෙවීම

15. “bubble sort හි ලැයිස්තුව හරහා එක් සම්පූර්ණ ගමන” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ඇල්ගොරිතම කාර්යක්ෂමතාව
B. අනුක්‍රමික සෙවීම
C. Pass
D. Swapped flag
E. වර්ග කිරීම

16. “අයිතම දෙකක ස්ථාන මාරු කිරීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. හුවමාරුව
B. Bubble sort
C. සෙවීම
D. ඉලක්කය
E. ආරෝහණ පිළිවෙළ

17. “හුවමාරු නැති pass එකකදී නවත්වන Boolean” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. අනුක්‍රමික සෙවීම
B. −1 ආපසු
C. Pass
D. සැසඳීම
E. Swapped flag

18. “කුඩාම සිට විශාලතම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ඇල්ගොරිතම කාර්යක්ෂමතාව
B. වර්ග කිරීම
C. ඉලක්කය
D. ආරෝහණ පිළිවෙළ
E. හුවමාරුව

19. “ඉලක්කය හමු නොවූ බවට සංඥාව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. සැසඳීම
B. Swapped flag
C. සෙවීම
D. Bubble sort
E. −1 ආපසු

20. “දත්ත වැඩි වන විට වැඩ වැඩි වන ආකාරය” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. අනුක්‍රමික සෙවීම
B. ආරෝහණ පිළිවෙළ
C. වර්ග කිරීම
D. ඇල්ගොරිතම කාර්යක්ෂමතාව
E. Pass

ව්‍යුහගත ප්‍රශ්න

1. සෙවීමේ ශ්‍රිතය. *(ලකුණු 3)*

> for; if; return.

2. 23 සහ 99. *(ලකුණු 2)*

> 2, 3; −1, 6.

3. pass පෙන්වන්න. *(ලකුණු 3)*

> pass 3.

4. swapped flag. *(ලකුණු 3)*

> කලින් නවතී.

5. හොඳම / නරකම. *(ලකුණු 2)*

> 1; n.

සාරාංශය

- සෙවීම පිළිවෙළට පරීක්ෂා කරයි.
- index හෝ −1 දෙයි.
- Bubble sort යාබද හුවමාරු කරයි.
- සෑම pass එකක්ම විශාලතම අගට.
- හුවමාරු නැති pass නවත්වයි.

MCQ පිළිතුරු – 9.13

1. **A** – 0 සිට.
2. **D** – 34, 7, 23.
3. **C** – හමු නොවේ.
4. **B** – 5 ගමන් කරයි.
5. **A** – 3 + 1.
6. **E** – යාබද.
7. **D** – කලින් නවතී.
8. **C** – n − 1.
9. **E** – සෙවීම – අගයක් ලැයිස්තුවේ කොතැනද යන්න සොයා ගැනීම.
10. **B** – අනුක්‍රමික සෙවීම – මුල සිට එකින් එක පරීක්ෂා කිරීම.
11. **A** – ඉලක්කය – සොයන අගය.
12. **C** – සැසඳීම – අයිතමයක් ඉලක්කය හෝ අසල්වැසියා සමඟ පරීක්ෂාව.
13. **B** – වර්ග කිරීම – ආරෝහණ හෝ අවරෝහණ පිළිවෙළට සැකසීම.
14. **B** – Bubble sort – වැරදි පිළිවෙළේ යාබද අයිතම හුවමාරු කරන sort.
15. **C** – Pass – bubble sort හි ලැයිස්තුව හරහා එක් සම්පූර්ණ ගමන.
16. **A** – හුවමාරුව – අයිතම දෙකක ස්ථාන මාරු කිරීම.
17. **E** – Swapped flag – හුවමාරු නැති pass එකකදී නවත්වන Boolean.
18. **D** – ආරෝහණ පිළිවෙළ – කුඩාම සිට විශාලතම.
19. **E** – −1 ආපසු – ඉලක්කය හමු නොවූ බවට සංඥාව.
20. **D** – ඇල්ගොරිතම කාර්යක්ෂමතාව – දත්ත වැඩි වන විට වැඩ වැඩි වන ආකාරය.

© 2026 schoolict.net (Ruwan Mangala Suraweera). සියලු හිමිකම් ඇවිරිණි.