

උසස් පෙළ තොරතුරු හා සන්නිවේදන තාක්ෂණය  
Advance Level Information & Communication Technology

"ජර්ගණක ක්‍රමලේඛණය විලිඛිත වෙනස්කම් අධ්‍යයනය කළ නොකළ වියය  
ධාරාවන් ඛණ්ඩය වෙත ප්‍රශ්නෝත්තර වාලාව නිර්මාණය කර ඇත"

# ජර්ගණක ක්‍රමලේඛණය හා Python

බහුවරණ ප්‍රශ්න සහ පිළිතුරු  
500 ක් කවෙහිත්

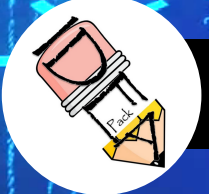
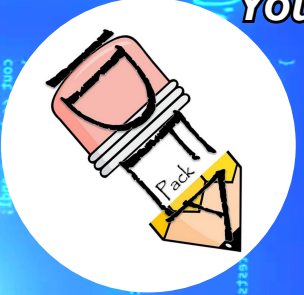
# ජර්ගණක ක්‍රමලේඛණය හා Python

බහුවරණ ප්‍රශ්න සහ පිළිතුරු  
500 ක් කවෙහිත්



Subscribe to our  
**YouTube Channel**  
[YouTube.com / ideaPack.lk](https://www.youtube.com/ideaPack.lk)

Tel : 076 3049 362



ideaPack PRODUCTION

උසස් පෙළ තොරතුරු හා සන්නිවේදන තාක්ෂණය  
පරිගණක ක්‍රමලේඛණය  
හා පයිතන් බහුවරණ  
ප්‍රශ්න හා පිළිතුරු

Advance Level Information and Communication Technology  
Programming and Python MCQ

ප්‍රශ්න 500ක් හා සියලුම පිළිතුරු සමගින්  
With 500 MCQs

කතෘ - K.A.D බිම්සර සඳරුවන්

උසස් පෙළ තොරතුරු හා සන්නිවේදන තාක්ෂණය  
 ක්‍රමලේඛණය හා පයිතන් බහුවරණ ප්‍රශ්න හා පිළිතුරු

පළමු මුද්‍රණය - 2019 අප්‍රේල්

කතෘගේ අවසරයකින් තොරව මෙහි අන්තර්ගත කිසිවක් අන්තර්ජාලය හෝ වෙන යම් ක්‍රමයකින් සම්ප්‍රේශණය කිරීම, උපුටා පළ කිරීම හෝ එසේ පළ කිරීමට හැකිවන ලෙස ගබඩා කර තබා ගැනීම සපුරා තහනම්.

කවර නිර්මාණය - වමිදු දයාන් පෙරේරා.  
 කඩුවෙල

ප්‍රකාශනය - ideaPack lk Productions  
 153/D මාලිගාගොඩුල්ල පාර,  
 උඩුමුල්ල මුල්ලේරියාව.

YouTube Channel: [ideaPack lk](#)

E-mail: [ideapacklk@gmail.com](mailto:ideapacklk@gmail.com)

Voice: 076 304 93 62 / 011 207 69 01



# ideaPack lk Productions

## කතෘ ප්‍රකාශය

උසස් පෙළ තොරතුරු තාක්ෂණය විශය නිර්දේශයේ පවතින, සිසුන් ගේ මතය අනුව අපහසුම පාඩමක් ලෙස ක්‍රමලේඛණය හා පයිතන් සැලකේ. මෙම මතය යම් තරමකින් හෝ අවම කිරීමට ගත් උත්සහයක විශිෂ්ට ප්‍රතිඵලයක් ලෙස මෙම පොත සැලකිය හැකිය.

මෙම පොතෙහි විශේෂත්වය වනුයේ සෑම ප්‍රශ්නයක්ම පාහේ මා විසින්ම නිපදවූ නැවුම් හා කිසිවිට නොදැකපු ප්‍රශ්න 500 කින් සමන්විත වීමයි. තවද මෙහි කිසිදු පසුගිය උසස් පෙළ ප්‍රශ්න පත්‍රවල තිබූ ප්‍රශ්න ඇතුළත්ව නොමැත. මක්නිසාද යත් එම ප්‍රශ්න වලට ඔබ පිළිතුරු සපයා ඇති බව නො අනුමානයයි.

සෑම ප්‍රශ්නයකට පාහේ තනිව පිළිතුරු සැපයීමට උත්සහ කරන්න. නිසැකවම ඔබ නොදන්නා ප්‍රශ්න කිසිපයක් හෝ මෙහි අඩංගු විය හැකිය. එබැවින් ඔබ නොදන්නා දෑ අන්තර්ජාලය මගින් සොයාගන්න.

ස්තූතියි...

ideaPack LK

## උසස් පෙළ තොරතුරු හා සන්නිවේදන තාක්ෂණය

### පරිගණක ක්‍රමලේඛණය හා පයිතන් බහුවරණ ප්‍රශ්න

1. පයිතන් පරිගණක ක්‍රමලේඛණ භාෂාව නිපදවන ලද්දේ කවුරුන් විසින් ද?
  - i) **Ada Lovelace**
  - ii) **Charles Babbage**
  - iii) **Guido Van Rossum**
  - iv) **Robert E. Kahn**
  - v) **Steve Jobs**



Python Logo

2. **Hyper Text Markup Language** හා සසඳන කල පයිතන් භාෂාවේ පවතින වෙනස වන්නේ,
  - i) පයිතන් හි **Comments** භාවිතයට හැකියාව පවතී.
  - ii) **Hyper Text Markup Language** යනු පරිගණක ක්‍රමලේඛණ භාෂාවකි.
  - iii) **Hyper Text Markup** භාෂාව වෙබ් පිටු නිපදවීම සඳහා ද පයිතන් භාෂාව පරිගණක ක්‍රමලේඛණ භාෂාවකි.
  - iv) පයිතන් මගින් ක්‍රමලේඛණය අනෙකුත් ක්‍රමලේඛණ භාෂාවලට සාපේක්ෂ ඉතා අපහසු ය.
  - v) පයිතන් වෙබ් පිටු සඳහා ද යම් දුරකට භාවිතා කල හැකිය.

3. ඇල්ගොරිතමයක් (**Algorithm**) යනු,
  - i) ක්‍රියාවක් සිදුකර ගන්නා ආකාරය අඩංගු වන සටහනකි.
  - ii) ක්‍රියාවක් සිදුකර ගන්නා ආකාරය පියවර වශයෙන් අඩංගු කරන ලද සටහනකි.
  - iii) ක්‍රියාවක් සිදුකර ගන්නා ආකාරය පියවර අනුපිළිවෙලක් සහිත හෝ රහිතව අඩංගු කරන ලද සටහනකි.
  - iv) ක්‍රියාවක් සිදුකර ගන්නා ආකාරය පියවර අනුපිළිවෙලක් සහිතව අඩංගු කරන ලද සටහනකි.
  - v) ඉහත කිසිවක් නොවේ.

4. ඇල්ගොරිතමයක් (**Algorithm**) සරලව පෙන්වා දැක්වීම සඳහා යොදාගනු ලබන ක්‍රමවේද වන්නේ,
  - i) **Python Codes and Java Codes**
  - ii) **Data Flow Diagram Level 1 and Data Flow Diagram Level 2 (DFD)**
  - iii) **ER Diagram and Flow Chart**
  - iv) **Flow Chart and Pseudocodes.**
  - v) ඉහත සියල්ලම

5. මෘදුකාංග පද්ධතියක් නිපදවීමේ දී එය කුඩා කොටස් වලට බෙදා කේතනය සිදුකිරීම හඳුන්වන්නේ,
  - i) මොඩියුලකරණය (**Module**)
  - ii) කේත බෙදුම (**Code Sharing**)
  - iii) වරණ පාලනය (**Selections**)
  - iv) ඉහත **i)** හා **ii)** පමණි.
  - v) ඉහත කිසිවක් නොවේ.

6. පරිගණක කේතයක පවතින වැරදි හා ඒවා නිවැරදි කිරීම ක්ෂේත්‍රයේ දී හඳුන්වනු ලබන්නේ,
  - i) **Bug and Bug Correcting**
  - ii) **Error and Error Fixing**
  - iii) **Faults and Correct the Wrong**
  - iv) **Bugs and Bugs Fixing**
  - v) **Errors and Errol**

7. පහත ප්‍රකාශ අතරින් නිවැරදි ප්‍රකාශය තෝරන්න.

- a. පළමු පරම්පරාවේ පරිගණක භාෂාවලදී හා දෙවන පරම්පරාවේ භාෂාවලදී භාෂා පරිවර්තක භාවිතා නොවීය.
  - b. පයිතන් යනු දෙවන පරම්පරාවේ භාෂාවකි.
  - c. පළමු පරම්පරාවේ භාෂාවලදී **Language Translators** භාවිතා නොවූනේ එය කෙලින්ම ප්‍රභව කේතයෙන්ම ක්‍රමලේඛණය සිදුකළ නිසා ය.
- i) a පමණි    ii) a , b පමණි    iii) c පමණි.    iv) ඉහත සියල්ලම    v) ඉහත කිසිවක් නොවේ

8. Python හි 1.23 යන්න ගබඩා කිරීම සඳහා භාවිත කල යුතු දත්ත ප්‍රරූපය (Variable Type) කුමක්ද?

- i) Float                      ii) int                      iii) String                      iv) Char                      v) Long

9. මේ අතරින් නිවැරදි විචල්‍ය නාමය කුමක්ද?

- i) Name@                      ii) \_Name                      iii) %name                      iv) Name-First                      v) First\_Name

10. පහත කේතයේ ප්‍රතිදානය, පිළිතුරු අතරින් කුමක්ද?

```
BEGIN
Print("Hi Man")
a = 0
while a >= 5 Do
    Print("Hi Man")
End while
END
```

- i) Hi Man ලෙස ප්‍රතිදානය නොකරයි.
- ii) Hi Man එක් වරක් ප්‍රතිදානය වේ.
- iii) Hi Man 2 වරක් ප්‍රතිදානය වේ.
- iv) Hi Man 5 වරක් ප්‍රතිදානය වේ.
- v) Hi Man 7 වරක් ප්‍රතිදානය වේ.

11. ගැලීම් සටහන් භාවිතයේ ප්‍රතිදාන, ක්‍රියායන, ආදාන පෙන්වනු ලබන්නේ පිළිවෙලින්, [ input , output , process ]

- i) ත්‍රිපිසියම, කොටුව, ත්‍රිපිසියම
- ii) සමානුපාතය, සමවතුරප්‍රය, සමානුපාතය
- iii) රවුම, සමවතුරප්‍රය, රවුම
- iv) ත්‍රිකෝණය, රවුම, කොටුව
- v) සම්මතයක් නොපවතී.

12. ක්‍රමලේඛණ භාෂා පරිවර්තකයක අවශ්‍යතාවය වඩාත් හොඳින් පැහැදිලි කර ඇත්තේ,

- i) ඉංග්‍රීසි වාක්‍යන් පරිගණකයට **Select** කර ඇති භාෂාවට හැරවීම.
- ii) ඉංග්‍රීසි හා සංකේත මඟින් ක්‍රමලේඛිත කේතය පරිගණකයට හඳුනා ගන්නා ප්‍රභව කේතය බවට හැරවීම.
- iii) ප්‍රභව කේතය යාන්ත්‍රික භාෂාව බවට හැරවීම.
- iv) එක් පරිවර්තනයකදී ඉහත සියල්ලම සිදුවේ.
- v) යාන්ත්‍රික කේතයේ ඇති වැරදි හඳුනාගෙන එය පරිශීලකයාට ඔහු දන්නා භාෂාවෙන් දැන්වීම.

13. පහත ප්‍රකාශ අතරින් නිවැරදි වන්නේ,

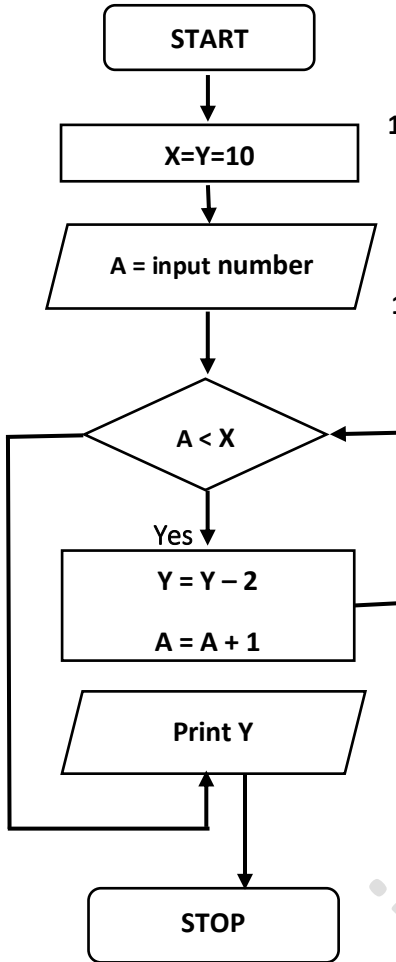
- a. ඇසෙම්බ්ලර් ක්‍රමලේඛණ භාෂාවහි භාෂා පරිවර්තකය ඇසෙම්බ්ලි පරිවර්තකය වේ.
- b. සම්පාදක මඟින් සම්පූර්ණ කේතයම එකවර යාන්ත්‍රික භාෂාවට හරවයි.
- c. අර්ථවිනාශක මඟින් ජේලියෙන් ජේලිය කේතය ප්‍රභව කේතය බවට හරවයි.

- i) a පමණි    ii) a හා b පමණි    iii) b හා c පමණි    iv) b පමණි    v) ඉහත කිසිවක් නොවේ.

14. පයිතන් කේතන මෘදුකාංගයේදී (Python IDE) භාවිතාවන භාෂා පරිවර්තක වර්ගය / වර්ග මොනවාද?

- i) Assembler    ii) Compiler    iii) Interpreter    iv) Compiler and Interpreter    v) All

>>> පහත ගැලීම් සටහන සලකා 15, 16, 17 ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.



15. පහත පිළිතුරු අතරින් මෙම ඇල්ගොරිතමයේ ප්‍රතිදානයක් නොවන්නේ,

- i) 0    ii) -2    iii) 9    iv) 2    v) 6

16. මෙම ඇල්ගොරිතමය සඳහා 1 ආදානය කල විට ප්‍රතිදානය වන්නේ,

- i) -8    ii) -9    iii) -10    iv) 8    v) ඉහත කිසිවක් නොවේ.

17. ඉහත කේතයට අදාළ ව්‍යාජ කේතය වන්නේ,

- i) 

```
begin
y = 10
x = y
A = input number
until A >= X do
  Y = Y - 2
  A = A + 1
end until
print Y
end
```
  - ii) 

```
begin
x = y
y = 10
A = input number
while A < X do
  Y = Y - 2
  A = A + 1
end while
print Y
end
```
  - iii) 

```
begin
y = 10
x = y
A = input number
while A < X do
  Y = Y - 2
  A = A + 1
end while
print Y
end
```
  - iv) 

```
begin
x = y
y = 10
A = input number
while A < X do
  Y = Y + 2
  A = A - 1
end while
print Y
end
```
- v) iii හා iv පමණි.

18. පයිතන් භාෂාව සම්බන්ධව වඩාත්ම නිවැරදි නොවන්නේ,

- i) පයිතන් යනු අනෙක් ක්‍රමලේඛණ භාෂාවලට සාපේක්ෂව ඉගෙන ගැනීමට ලෙහෙසි නොවන භාෂාවකි.
- ii) පයිතන් භාෂාවේදී ක්‍රමලේඛණ ධාවනය සඳහා භාෂා පරිවර්තකයක් අත්‍යවශ්‍යම වේ.
- iii) Java, C, Prolog, වැනි ක්‍රමලේඛ භාෂා නිසා පයිතන් භාෂාව අද වනවිට භාවිතයේ නැති තරම් ය.
- iv) පයිතන් යනු ඉහළ මට්ටමේ පරිගණක ක්‍රමලේඛණ භාෂාවකි.
- v) පයිතන් යනු අර්ථවිනාශක භාෂාවකි.

19. Floor Division ලෙස හඳුන්වනු ලබන්නේ,

- i) /    ii) //    iii) %    iv) &&    v) ~



20. X නම් වූ පරිගණක භාෂාවහි කේත Y නම් වූ පරිවර්තකය මඟින් යාන්ත්‍රික භාෂාවට පරිවර්තනය කරයි. X හා Y පිළිවෙළින් දැක්වෙන්නේ,

- i) පයිතන්, ඇසෙම්බ්ලර්
- ii) පයිතන්, පයිටරාන්ස(ර්)
- iii) ඇසෙම්බ්ලර්, ඇසෙම්බ්ලි
- iv) Java, සම්පාදක
- v) ඇසෙම්බ්ලි, ඇසෙම්බ්ලර්

21. පහත ප්‍රකාශ අතරින් නිවැරදි වගන්ති තෝරන්න.

- a. යාන්ත්‍රිකව ලියන ලද කේතයක් අනෙකුත් ක්‍රමලේඛණ භාෂා පරම්පරාවල භාෂා වලින් ලියන ලද කේත වලට සාපේක්ෂව ඉතා ඉක්මනින් ක්‍රියාත්මක වේ.
  - b. පළමු පරම්පරාවේ ක්‍රමලේඛ භාෂාවල භාෂා පරිවර්තක භාවිතා නොවීය.
  - c. ඇසෙම්බ්ලි යනු පළමු පරම්පරාවේ ක්‍රමලේඛණ භාෂාවකි.
  - d. හතර වෙනි පරම්පරාවේ භාෂා සෑම පරිගණකයකම සෘජුව ධාවනය කල හැකිය.
- i) a පමණි.    ii) b පමණි.    iii) a , b පමණි.    iv) a , d හා b පමණි.    v) a හා c පමණි.

22. විවරණ යනු (comments) කේතකරණයේදී භාවිතා කරන අංගයකි. ඒවා සම්බන්ධයෙන් පහත ප්‍රකාශ අතරින් වඩාත්ම නිවැරදි ප්‍රකාශය තෝරන්න.

- i) විවරණ පරිවර්තක මඟින් යාන්ත්‍රික භාෂාව මඟින් පරිවර්තනය කලත් ධාවනය නොකරයි.
- ii) විවරණ පරිවර්තක මඟින් යාන්ත්‍රික භාෂාව මඟින් පරිවර්තනය වී ධාවනය වුවත් එය පරිශීලකට නොපෙනේ.
- iii) විවරණ යෙදීම අත්‍යවශ්‍ය නොවන මුත් එයින් කේතනය කරන අවස්ථාවේදී පහසුවක් ඇතිකරයි.
- iv) විවරණ මතු ප්‍රයෝජන සඳහා හොඳ ආයෝජනයකි.
- v) සමහර පරිගණක භාෂාවල {if ; } ලෙස විවරණ ආරම්භ වේ.

>>> ඇල්ගොරිතමයෙන් 23, 24, 25, 26 ප්‍රශ්න වලට පිළිතුරු සපයන්න. පහත කේතයේ Stop විදානයෙන් පසු වැඩසටහන සම්පූර්ණයෙන්ම අවසන් වේ. එවිට ඊට පසුව ඇති කේත ධාවන නොවේ. 1 හෝ ඊට අඩු අගයන් ආදානය කොකරන බව සලකන්න.

```

Count = 2
Val = input integer value 2 to 100
While ( Count <= Val ) do
  IF ( Val % Count = 0 and Val != Count ) then
    Print " NO "
    Stop (වැඩසටහන අවසන් වේ.)
  Count = Count + 1
Print "YES"
Stop (වැඩසටහන අවසන් වේ.)

```

- 23. " 2 " ආදානය කලපසු ප්‍රතිදානය වන්නේ,  
i) NO ii) YES iii) YES හා NO iv) NO හා YES v) -
- 24. " 11 " ආදානය කලපසු ප්‍රතිදානය වන්නේ,  
i) NO ii) YES iii) YES හා NO iv) NO හා YES v) -
- 25. Val හි විචල්‍ය වර්ගය වන්නේ,  
i) Int ii) String iii) Char iv) Float v) Long

26. ඉහත ඇල්ගොරිතමය මඟින් ආදානය කරන ලද අගය සඳහා YES ලෙස ප්‍රතිදානය කළ හොත් එය,  
i) ඉරට්ටේ අගයකි. ii) ඔත්තේ අගයකි. iii) ප්‍රථමක සංඛ්‍යාවකි. iv) කිව නොහැකිය. v) දශම අගයකි.

27. පයිතන් භාෂාවේ විචල්‍ය නාමයන් අතරින් නිවැරදි වන්නේ,  
i) YoYo\_@ ii) yoyo-2108 iii) yOYO\_2018 iv) Yoyo. v) \_2yoyo18

28. පයිතන් හී ධාවනය කල විට දෝෂ පණිවිඩ ලබා නොදෙන පිළිතුර වන්නේ (පහත දක්වා ඇත්තේ කේතයේ පළමු ඡේදිය යි),

- i) >>> a = 10.
- ii) >>> a == 10
- iii) >>> # .# (a = 10.)
- iv) >>> a = b = 10 = c
- v) i හා iii

29. පහත ක්‍රමලේඛණ භාෂා අතරින් ඉහළ මට්ටමේ භාෂාව / භාෂා වන්නේ,

- i) Python
- ii) Java
- iii) Prolog
- iv) i , ii
- v) i, ii, iii

30. පහත ප්‍රකාශ සලකා නිවැරදි වගන්තිය / වගන්ති තෝරන්න.

- a. ශබ්දාර්ථ දෝෂ (Semantic Errors) සහිත කේත සම්පූර්ණයෙන්ම ධාවනය විය හැකිය
- b. කාරක දෝෂ (Syntax Errors) සහිත කේත මූල සිට අභ්‍යන්තර කිසිවිටක ධාවනය නොවේ.
- c. ශබ්දාර්ථ දෝෂ නිවැරදි ආදානයන්ට වැරදි ප්‍රතිදාන දීමට හේතු විය හැකිය.

- i) a පමණි.
- ii) c පමණි.
- iii) b , c පමණි.
- iv) a , b පමණි.
- v) ඉහත සියල්ලම.

31. පයිතන් කේතයක් පරිගණකයේ Save කළ විට එහි ගොනු දිගු නාමය වන්නේ,

- i) .exe
- ii) .pyt
- iii) .python
- iv) .py
- v) .pyn

32. සම්පූර්ණ ප්‍රභව කේතය එකවර විශය කේතය බවට හරවන හා සම්පූර්ණ විශය කේතය ජේලියෙන් ජේලිය විශය කේතය බවට හරවනු ලබන්නේ පිළිවලින්,

- i) Compiler Only
- ii) Compiler and Interpreter
- iii) Compiler and Assembler
- iv) Assembler and Interpreter
- v) None Above

33. පයිතන් භාෂාවේ පහත ප්‍රකාශන වල සඳහන් විචල්‍යයන් නිවැරදි පිළිවලට දක්වා ඇති පිළිතුර තෝරන්න.

- a. X = { 1,2,3,4,5}
  - b. Y = 2 , 3
  - c. P,Q = 1.2 , 1.4
- i) Dictionary , Tuple , Tuple
  - ii) Dictionary, Int, Float
  - iii) Dictionary, Tuple, Float
  - iv) Set, Tuple, Float
  - v) Dictionary or Set, Tuple, Float

34. Mutable හා Immutable ලෙස විචල්‍ය වර්ග කරයි. එම වර්ගවලට අදාළ විචල්‍ය වර්ග පිළිවෙලින් දැක්වෙන්නේ,

- i) Int, String, Float හා List, Dictionary Set
- ii) List, Dictionary, Set හා Int, String, Float
- iii) List, Dictionary හා Set, Int, String, Float
- iv) ඉහත ලෙස වර්ගීකරණයක් නැත.
- v) Int, String, Set හා Float List, Dictionary

35. විචල්‍යයක් නිර්මාණය කිරීමට පෙර සැලකිලිමත් විය යුතු කරුණු ලෙස පහත කරුණු දක්වා ඇත. ඒවායින් නිවැරදි පිළිතුරු තෝරන්න.

- a. විචල්‍යයේ ගබඩාකිරීමට බලාපොරොත්තු වන දත්ත වර්ගය කුමක්ද යන වග.
  - b. විචල්‍යයේ ගබඩාකිරීමට බලාපොරොත්තු වන දත්තයේ ප්‍රමාණය පිළිබඳව.
  - c. විචල්‍යයේ අලංකාරය පිළිබඳව.
- i) a පමණි.
  - ii) b පමණි.
  - iii) a , b පමණි.
  - iv) c පමණි.
  - v) කිසිවක් නොවේ.

36. පයිතන් අර්ථවිනායකය මගින් පහත කේතය ධාවනය කළ පසු ලැබෙන ප්‍රතිදානය තෝරන්න.

```
>>> a = { 3,2,1,0 }
>>> print(a)
```

- i) {3,2,1,0}
- ii) "{3,2,1,0}"
- iii) (0,1,2,3)
- iv) {0,1,2,3}
- v) error

37. පයිතන් අර්ථවිනායකය මගින් පහත කේතය ධාවනය කළ පසු ලැබෙන ප්‍රතිදානය තෝරන්න.

```
>>> "YOYO"
```

- i) "YOYO"
- ii) 'YOYO'
- iii) YOYO
- iv) Error
- v) ""YOYO""

38. පයිතන් අර්ථවිනායකය (interpreter) මඟින් පහත කේතය ධාවනය කළ පසු ලැබෙන ප්‍රතිදානය තෝරන්න.

```
>>> print(" YOYO ")
```

- i) " YOYO "
- ii) ' YOYO '
- iii) YOYO
- iv) Error
- v) " " YOYO " "

39. පයිතන් අර්ථවිනායකය මඟින් පහත කේතය ධාවනය කළ පසු ලැබෙන ප්‍රතිදානය තෝරන්න.

```
>>> print(" YOYO ")
```

- i) " YOYO "
- ii) ' YOYO '
- iii) YOYO
- iv) Error
- v) " " YOYO " "

40. පයිතන් අර්ථවිනායකය මඟින් පහත කේතය ධාවනය කළ පසු ලැබෙන ප්‍රතිදානය තෝරන්න.

```
>>> print(" YOYO ")
```

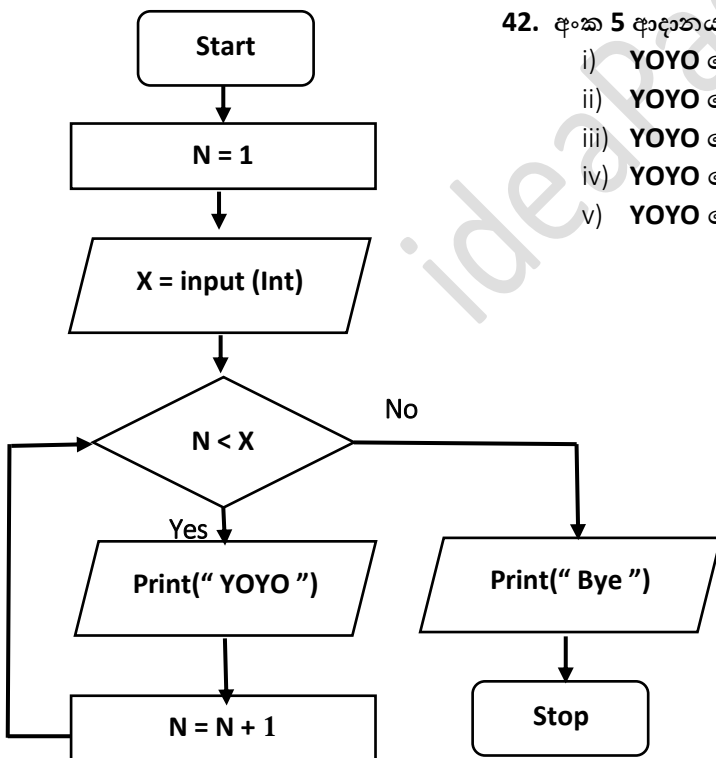
- i) " YOYO "
- ii) ' YOYO '
- iii) YOYO
- iv) Error
- v) " " YOYO " "

41. පහත අතරින් නිවැරදි ප්‍රකාශය / ප්‍රකාශ තෝරන්න.

- a. List එකක් යනු [ ] යන කොටු වරහන් වලින් වට වී අනුපිළිවෙලකට සකස් වී ඇති දත්ත සමූහයකි.
- b. Set එකක් යනු { } යන සහල වරහන් වලින් වට වී අනුපිළිවෙලකට සකස් වී ඇති දත්ත සමූහයකි.
- c. Set , Tuple වලට නව දත්ත ඇතුළු කල නොහැකිය. ඉවත් කිරීමටද නොහැකිය.
- d. Dictionary යනු Mutable විවලය වවර්ගයකි.

- i) a, b පමණි.
- ii) a, b, c පමණි.
- iii) a, c පමණි.
- iv) a, d පමණි.
- v) b, c, d පමණි.

>>> පහත ගැලීම් සටහන සලකා 42, 43, 44 ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.



42. අංක 5 ආදානය කළ විට,

- i) YOYO ලෙස 5 වරක්ද Bye ලෙස එක් වරක්
- ii) YOYO ලෙස 6 වරක්ද Bye ලෙස එක් වරක්
- iii) YOYO ලෙස එකවරක් හා Bye ලෙස එක් වරක්
- iv) YOYO ලෙස 4 වරක්ද Bye ලෙස එක් වරක්
- v) YOYO ලෙස 4 වරක්ද Bye ලෙස 4 වරක්

43. ආදානයේ X අගය හා ප්‍රතිදාන පාඨ ගණන Y

අතර සම්බන්ධතාව වන්නේ,

- i) X = Y - 1
- ii) X = Y - 2
- iii) X = Y -- 1
- iv) X + 1 = Y -- 1
- v) සම්බන්ධයක් නොමැත.

44. 0 ආදානය කල විට ප්‍රතිදානය වන්නේ,

- i) YOYO ලෙස එකවරක් හා Bye ලෙස එකවරක්
- ii) YOYO ලෙස ප්‍රතිදානය නොවන අතර Bye ලෙස එකවරක්

iii) ප්‍රතිදානයන් නැත

iv) **Bye** පමණි.

v) **Error**

>>> පහත කේතය සලකා **45 – 47** ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
total = int(0.9999)
x = 0
while x >= 0 :
    x = int (input(" Enter a Value : "))
    total = total + x
print(total)
```

**45.** total හි ගබඩාවන අගය වන්නේ,

- i) **0**                      ii) **1**                      iii) **0.90**                      iv) **0.9999**                      v) **Error**

**46.** ඉහත කේතයේ වැරදි ඇත්නම් ඒවද නිවැරදි කල පසු පිළිවෙලින් **10, 12, 11, - 10** ආදානය කරන ලදී. එවිට ප්‍රතිදානය වන්නේ,

- i) **32**                      ii) **33**                      iii) **-33**                      iv) **23**                      v) කිසිවක් නොවේ.

**47.** ආදානය කරනු ලබන ආදාන **5**ක එකතුව ලබා ගැනීමට කේතයේ සිදු කල යුතු වෙනස්කම වන්නේ,

- i) **while x <= 5** පමණි.                      ii) **while x >= 5** පමණි.                      iii) **while x < 5** පමණි.
- iv) **while x != 5** පමණි.                      v) ඉහත කිසිවක් නොවේ.

**48.** පයිතන් හි කාරක දෝෂ රහිත කේතය වන්නේ,

- i) **a = int(10. )**                      ii) **int = 10**                      iii) **a = int("YOYO")**                      iv) **i හා ii** පමණි.                      v) කිසිවක් නොවේ.

>>> **49** සිට **100** දක්වා ඇති ප්‍රශ්න සඳහා, එය පයිතන් අර්ථවිනයාසයේ ධාවනය කලවිට ලැබෙන ප්‍රතිඵලය සොයන්න.

**49.** **2 / 2** ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) **1**                      ii) **1.0**                      iii) **1 හෝ 1.0**                      iv) දෝෂ සහිතයි.                      v) **1.00**

**50.** **2 // 2** ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) **1**                      ii) **1.0**                      iii) **1 හෝ 1.0**                      iv) දෝෂ සහිතයි.                      v) **1.00**

**51.** **999 % 10** ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) **99**                      ii) **99.9**                      iii) **9**                      iv) **99.0**                      v) **9.0**

**52.** **(2 \*\* 2+1) \*\* 3** ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) **256**                      ii) **12**                      iii) **64**                      iv) **512**                      v) **125**

**53.** **2 \* 2 \*\* 3** ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) **16**                      ii) **64**                      iii) **125**                      iv) **512**                      v) **256**

**54.** **3 % 10 \* 3** ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) **0**                      ii) **9**                      iii) **0.99**                      iv) **9.0**                      v) **256**

**55.** **31 // 2 + 3 \*\* 2 + 10** ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) **32**                      ii) **22**                      iii) **64**                      iv) **34**                      v) **55**

**56.** **2 + 3 \*\* 3 // 4 \* (9 % 3 + 5 / 2 ) - - 1** ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) **18**                      ii) **18.0**                      iii) **29.0**                      iv) **29**                      v) **55**

57.  $10 -- 1 + 10 --- 1$  ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) 20                      ii) 22                      iii) 18                      iv) 21                      v) Error

58.  $True + 1$  ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) 2                      ii) True                      iii) 1                      iv) Empty                      v) Error

59.  $False + 1 + True$  ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) True                      ii) 2                      iii) Error                      iv) 1                      v) Empty

60.  $not + 1$  ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) 2                      ii) 1                      iii) 0                      iv) False                      v) Error

61.  $0 ** 0$  ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) 1                      ii) 0                      iii) Error                      iv) Infinity                      v) Empty

62.  $0 ** 2$  ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) 2                      ii) 0                      iii) Error                      iv) Infinity                      v) Empty

63.  $2 ** -1$  ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) 0                      ii) 2                      iii) 1/2                      iv) 0.5                      v) iii, iv

64.  $10 ** -2$  ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) 0                      ii) 0.1                      iii) 0.01                      iv) 0.001                      v) Error

65.  $(-2 ** 4)$  හා  $(2 * 2 * 2 * 2)$  වෙන වෙනම ධාවනය කල විට ප්‍රතිදානය ලිවෙලින් වන්නේ,

- i) -16 හා -16                      ii) 16 හා 16                      iii) Error හා 16                      iv) -16 හා 16                      v) 16 හා -16

66.  $-2 ** -2$  ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) -0.25                      ii) 0.25                      iii) -4                      iv) -1/4                      v) Error

67.  $(-2) ** 2$  හා  $-2 ** 2$  වෙන වෙනම ධාවනය කල විට ප්‍රතිදානය පිලිවෙලින් වන්නේ,

- i) 4 හා 4                      ii) 4 හා -4                      iii) -4 හා 4                      iv) -4 හා -4                      v) Error

68.  $1 is 1.0$  හා  $1 == 1.0$  වෙන වෙනම ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) True හා True                      ii) 1 හා 1                      iii) False හා True                      iv) 1 හා 0                      v) Error හා True

69.  $(-2 ** 3) == ((-2) ** 3)$  ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) True                      ii) 1                      iii) False                      iv) true                      v) Error

70.  $1.0 != 1$  ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) True                      ii) 1                      iii) False                      iv) true                      v) Error

71.  $1.0 is not 1$  ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) True                      ii) 1                      iii) False                      iv) true                      v) Error

72.  $\sim -9$  ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) 8                      ii) -8                      iii) 9                      iv) -9                      v) 10

73.  $(100.0 == 101-1)$  and  $(101.0 is 102.0 - 1)$  ධාවනය කල විට ප්‍රතිදානය වන්නේ,

- i) true                      ii) 1                      iii) False                      iv) True                      v) Error

74.  $(120.0 \neq 121-4)$  or  $(101.0 \text{ is } 102.0 - 1)$  ධාවනය කල විට ප්‍රතිදානය වන්නේ,  
 i) True                      ii) 1                      iii) False                      iv) true                      v) Error
75.  $(120.0 \neq 121-4)$  or  $(\text{not}(101.0 \text{ is } 102.0 - 1))$  ධාවනය කල විට ප්‍රතිදානය වන්නේ,  
 i) True                      ii) 1                      iii) False                      iv) true                      v) Error
76.  $(\text{True and True})$  හා  $(\text{True and } 0)$  වෙන වෙනම ධාවනය කල විට ප්‍රතිදානය පිළිවෙලින් වන්නේ,  
 i) True හා 0                      ii) True හා False                      iii) True හා Error                      iv) False හා True                      v) 1 හා 0
77.  $"A" > "B"$  හා  $"A" < "B"$  වෙන වෙනම ධාවනය කල විට ප්‍රතිදානය (output) පිළිවෙලින් වන්නේ,  
 i) True හා 0                      ii) True හා False                      iii) True හා Error                      iv) False හා True                      v) 1 හා 0
78.  $"X" > "Y"$  හා  $"x" < "y"$  වෙන වෙනම ධාවනය කල විට ප්‍රතිදානය (output) පිළිවෙලින් වන්නේ,  
 i) True හා 0                      ii) True හා False                      iii) True හා Error                      iv) False හා True                      v) 1 හා 0
79.  $"X" > "y"$  හා  $"a" < "A"$  වෙන වෙනම ධාවනය කල විට ප්‍රතිදානය (output) පිළිවෙලින් වන්නේ,  
 i) True හා 0                      ii) True හා False                      iii) True හා Error                      iv) False හා False                      v) 1 හා 0
80.  $"1" > "2"$  හා  $1 > 2$  වෙන වෙනම ධාවනය කල විට ප්‍රතිදානය (output) පිළිවෙලින් වන්නේ,  
 i) True හා True                      ii) False හා False                      iii) Error හා False                      iv) False හා True                      v) 0 හා 0
81.  $"Bimsara" > "Binsara"$  හා  $10.0 > 10$  වෙන වෙනම ධාවනය කල විට ප්‍රතිදානය (output) පිළිවෙලින් වන්නේ,  
 i) True හා True                      ii) False හා False                      iii) Error හා False                      iv) False හා True                      v) 0 හා 0
82.  $"YoYo" > "yOyO"$  ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ,  
 i) True                      ii) 0                      iii) Error                      iv) False                      v) 1
83.  $"B" > " "$  ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ, # B > Space  
 i) True                      ii) False                      iii) Error                      iv) 0                      v) 1
84.  $"Yo.Yo" > "YoYo"$  ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ,  
 i) True                      ii) False                      iii) Error                      iv) 0                      v) 1
85.  $10 \& 5$  ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ,  
 i) True                      ii) False                      iii) 1                      iv) 10                      v) 0
86.  $24 | 10$  ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ,  
 i) 24                      ii) 34                      iii) 26                      iv) 32                      v) 22
87.  $33 \wedge 45$  ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ,  
 i) 44                      ii) 90                      iii) 12                      iv) 26                      v) 67
88.  $10 | 3 \& 4$  ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ,  
 i) 8                      ii) 9                      iii) -10                      iv) 17                      v) 10
89.  $(10 | 3) \& 4$  ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ,  
 i) 8                      ii) 0                      iii) 10                      iv) 9                      v) -0
90.  $16 | 32 \& 8 \wedge 2$  ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ,  
 i) 8                      ii) 10                      iii) 18                      iv) 9                      v) -0

91.  $10 \ll 2$  ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ,  
 i) 2                      ii) True                      iii) False                      iv) 40                      v) 1000
92.  $33 \gg 3$  ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ,  
 i) 4                      ii) True                      iii) TrueTrue                      iv) 264                      v) 8
93.  $64 \ll -1$  ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ,  
 i) 32                      ii) 128                      iii) Error                      iv) 0                      v) -32
94.  $128 \ll 2 \gg 1 \ll 4$  ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ,  
 i) 2048                      ii) 4096                      iii) 8192                      iv) 4                      v) කිසිවක් නොවේ
95.  $128 \gg 2 \ll 1 \gg 4$  ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ,  
 i) 2048                      ii) 4096                      iii) 8192                      iv) 4                      v) කිසිවක් නොවේ.
96.  $0 / 2$  ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ,  
 i) 0                      ii) Error                      iii) Infinity                      iv) Empty                      v) කිසිවක් නොවේ.
97.  $2 / 0$  හා වෙන වෙන ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ,  
 i) 0                      ii) Error                      iii) Infinity:Error                      iv) Empty                      v) කිසිවක් නොවේ.
98.  $"b" / 2$  ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ,  
 i) 0                      ii) Error                      iii) 2 / 2                      iv) 1                      v) කිසිවක් නොවේ.
99.  $"A ." * 2$  ධාවනය කල විට ප්‍රතිදානය (output) වන්නේ,  
 i) Error                      ii) Double /A                      iii) AA.                      iv) A .A .                      v) 'A .A.'
100.  $"2" * 2$  හා  $2 * 2$  වෙන වෙනම ධාවනය කල විට ප්‍රතිදානය (output) පිළිවෙළින් වන්නේ,  
 i) 22 හා 4                      ii) 4 හා 4                      iii) Error හා 4                      iv) '22' හා 4                      v) "22" හා 4
101.  $a, b = 1,2$   
 $\text{print}(a + b)$  මෙහි ප්‍රතිදානය (output) වන්නේ,  
 i) "1 + 2"                      ii) "3"                      iii) 3                      iv) Error                      v) '12'
102.  $a, b = "12"$   
 $\text{print}(a)$   
 $\text{print}(b)$  මෙහි ප්‍රතිදානය (output) වන්නේ,  
 i) 1 හා 2                      ii) '1' හා '2'                      iii) Error                      iv) (12, 12)                      v) "12" හා "12"
103.  $a = 1,2,3$   
 $\text{print}(a)$  මෙහි ප්‍රතිදානය (output) වන්නේ,  
 i) Error                      ii) 123                      iii) [1,2,3]                      iv) '1','2','3'                      v) (1,2,3)
104.  $a = 1,2,3$   
 $\text{print}(\text{type}(a))$  මෙහි ප්‍රතිදානය (output) වන්නේ,  
 i) <class 'int'>                      ii) <class 'string'>                      iii) <class 'int1'>                      iv) <class 'tuple'>                      v) <class 'list'>
105.  $a = 1,$   
 $\text{print}(\text{type}(a))$  මෙහි ප්‍රතිදානය (output) වන්නේ,  
 i) <class 'int'>                      ii) <class 'string'>                      iii) <class 'int1'>                      iv) <class 'tuple'>                      v) <class 'list'>
106.  $a = '12'$   
 $\text{print}(\text{type}(a))$  මෙහි ප්‍රතිදානය (output) වන්නේ,  
 i) <class 'char'>                      ii) <class 'string'>                      iii) <class 'int'>                      iv) <class 'tuple'>                      v) <class 'list'>

107. `a = 'a,b,c'`,  
`print( type ( a ) )` මෙහි ප්‍රතිදානය (output) වන්නේ,  
 i) `<class 'char'>`    ii) `<class 'string'>`    iii) `<class 'int'>`    iv) `<class 'tuple'>`    v) `<class 'list'>`
108. `a , b = 1,2`  
`print(a)` මෙහි ප්‍රතිදානය (output) වන්නේ,  
 i) Error    ii) 1    iii) 12    iv) '12'    v) "12"
109. `a , b = 1,2,`  
`print(a)` මෙහි ප්‍රතිදානය (output) වන්නේ,  
 i) 1    ii) 2    iii) Error    iv) (1)    v) (1,2,0)
110. `a , b = (1,2,)`  
`print(b)` මෙහි ප්‍රතිදානය (output) වන්නේ,  
 i) 2    ii) 12,    iii) Error    iv) (2,0)    v) (1,2,0)
111. `a = b = 'a' , 'b'`  
`print(a)`  
`print(b)` මෙහි ප්‍රතිදානය (output) පිළිවලින් වන්නේ,  
 i) 'a' හා 'b'    ii) 'ab' හා 'ab'    iii) Error    iv) ('a', 'b') හා ('a', 'b')    v) "a" හා "b"
112. `t = [ 1 , 2 , " Yoyo " ]`  
`print( type ( t ) )`  
`print( type ( t [ 2 ] ) )` මෙහි ප්‍රතිදානය (output) පිළිවලින් වන්නේ,  
 i) tuple හා tuple In    ii) list හා list In    iii) list හා Error    iv) list හා String    v) list හා list
113. `x = [ 1 , 2 , 3 , ( 4 , [ 5 , 6 ] ) ]`  
`print( x [ 3 ] [ 1 ] )` මෙහි ප්‍රතිදානය (output) වන්නේ,  
 i) 5    ii) 6    iii) [5,6]    iv) Error    v) None
114. `x = { 3 , 5 , 2 , 3 , 6 }`  
`print( type(x) )` මෙහි ප්‍රතිදානය (output) වන්නේ,  
 i) Dictionary    ii) Set    iii) Tuple    iv) List    v) Error
115. `x = { 3 , 5 , 2 , 3 , 6 }`  
`print ( x )` මෙහි ප්‍රතිදානය (output) වන්නේ,  
 i) {3,5,2,3,6}    ii) [3,5,2,3,6]    iii) {2,3,3,5,6}    iv) {2,3,5,6}    v) Error
116. `x = { 1 : "a" , 2 : "b" , 3 : "c" }`  
`print ( type(x) )`  
`print ( x[1] )` මෙහි ප්‍රතිදානයන් පිළිවලින් (output) වන්නේ,  
 i) Set , 'a' , 'b'    ii) Dictionary , 'a'    iii) Dictionary , a , b  
 iv) Dictionary , '1'    v) Set , '2'
117. `y = { 1 : 10 , 2 : 20 , 3 : (35 , 40) }`  
`print ( y[1] + y[2][1] )` මෙහි ප්‍රතිදානය (output) වන්නේ,  
 i) 5    ii) 4    iii) 50    iv) 85    v) Error
118. `x = { "x" , "x" , "ax" , "c" }`  
`print ( x )` මෙහි ප්‍රතිදානය (output) වන්නේ,  
 i) {'a' , 'x' , 'c'}    ii) {'a' , 'ax' , 'c'}    iii) {'x' , 'c' , 'ax' , 'a'}  
 iv) {'c' , 'x' , 'ax'}    v) {'x' , 'a' , 'ax' , 'c'}



119. `y = { 1 : 10 , 2 : 20 , 3 : (35 , 40) }`

`print ( x[1] + x[3][1] )` මෙහි ප්‍රතිදානය (output) වන්නේ,

- i) 5
- ii) 4
- iii) 50
- iv) 85
- v) Error

120. `y = { 1 : 10 , 2 : 20 , 3 : (35 , 40) } ; print ( y[1] + y[3][1] )` මෙහි ප්‍රතිදානය වන්නේ,

- i) පයිතන් හී ' ; ' භාවිතයේ නැත.
- ii) 4
- iii) 50
- iv) 85
- v) Error

121. පහත ප්‍රකාශ අතරින් True ප්‍රතිදානය (output) කරන ප්‍රකාශය / ප්‍රකාශ තෝරන්න.

- a. `t = [ 1 , 2 , 3 ] ; b = [ 1 , 2 , 3 ] ; a == b`
- b. `[ 1 , 2 , 3 ] == [ 1 , 2 , ' 3 ' ]`
- c. `A = { 1 , 2 , 0 , 9 } != { 9 , 0 , 0 , 2 , 1 , 0 } ; print(A)`

- i) a පමණි.
- ii) b පමණි.
- iii) a , b පමණි.
- iv) a , c පමණි.
- v) කිසිවක් නොවේ.

122. පහත කේතයේ ප්‍රතිදානය (output) වන්නේ,

```
a , b = ( 56 , 89 )
c , d = { 20 , 10 }
print(a + c)
```

- i) 66
- ii) 144
- iii) 76
- iv) 99
- v) Error

>>> පහත ප්‍රකාශ මගින් 123 , 124 , 125 ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

- a. `x , y = 10 , 22`
- b. `[p , q] = 55 , 90`
- c. `( i and h ) == " YOYO " , " Productions "`
- d. `[ ( a , b ) , c , d ] = [ [ 2 , 2 ] , 8 , 0 ]`

123. මේ අතරින් වලංගු ප්‍රකාශ වන්නේ,

- i) a පමණි.
- ii) a , b පමණි.
- iii) a , b , d පමණි.
- iv) a , c පමණි.
- v) සියල්ල.

124. " c " ප්‍රකාශයේ පරිදි `print(i + h)` හි ප්‍රතිදානය වන්නේ,

- i) Error
- ii) YOYOproducts
- iii) ' YOYOproducts
- iv) " YOYOproducts "
- v) None

125. " d " ප්‍රකාශයේ පරිදි `print(a + d)` හි ප්‍රතිදානය වන්නේ,

- i) 2
- ii) '20'
- iii) Error
- iv) a not assign
- v) ඉහත කිසිවක් නොවේ.

126. පහත ප්‍රකාශ සලකන්න.

- a. `a = [ 1 , 2 , 3 , 4 , 5 ] [ 2 ]`
- b. `b = [ 1 , 2 , 3 , 4 , 5 , 6 ] + [ 2 ]`
- c. `c = [ 1 , 2 , 3 , 4 , 5 ] [ 1 : 4 ]`
- d. `d = [ 1 , 2 , 3 , 4 , 5 ] [ :: 2 ]`

ප්‍රකාශ වල ඇති විචල්‍ය ( variable ) වල ඇති ( a , b , c , d ) අගයන් පිළිවෙලින් ඇත්තේ,

- i) [ 1 , 2 , 3 , 4 , 5 , 2 ] , [ 1 , 2 , 3 , 4 , 5 , 2 ] , [ 1 , 2 , 3 , 4 ] , [ 3 ]
- ii) 3 , [ 1 , 2 , 3 , 4 , 5 , 6 , 2 ] , [ 2 , 3 , 4 ] , [ 1 , 3 , 5 ]
- iii) 3 , [ 1 , 3 , 4 , 5 , 2 ] , [ 2 , 3 , 4 , 5 ] , [ 2 , 4 ]
- iv) [ 3 ] , [ 1 , 2 , 3 , 4 , 5 , 6 , 2 ] , [ 2 , 3 , 4 , 5 ] , [ 1 , 3 , 5 ]
- v) Can not assign like this , [ 1 , 2 , 3 , 4 , 5 , 6 , 2 ] , [ 2 , 3 , 4 , 5 ] , [ 1 , 3 , 5 ]

127. පහත කේතයේ ප්‍රතිදානය වන්නේ,

```
a = ( 1 , 2 , 3 , 4 , " 5 " , 6 ) [ 2 ]
print ( type( a ) )
```

- i) <class 'char'>
- ii) <class 'string'>
- iii) <class 'int'>
- iv) <class 'tuple'>
- v) <class 'list'>

128. පහත ප්‍රකාශ සලකන්න.

- a. `w={'s','f','q','z'}[2] ; print(w)`
- b. `x={1:"SLTV", 2:"YOYO", 3:"USI"}[1] ; print(x)`
- c. `y={1:"SLTV", 2:"YOYO", 3:"USI"}["SLTV"] ; print(y)`
- d. `z={"x","x"} ; print(z)`

ඉහත ප්‍රකාශ වල ප්‍රතිදානයන් පිළිවෙළින් ඇති පිළිතුර වන්නේ,

- i) Error , 'SLTV' , Error , {'x'}
- ii) 'f' , 'SLTV' , 1 , {'x' , 'x'}
- iii) 'f' , 'SLTV' , Error , 'x'
- iv) Error , 'SLTV' , Error , {'x','x'}
- v) ඉහත කිසිවක් නොවේ.

129. පහත ප්‍රකාශ සලකන්න,

- a. `a={2.0} ; print( type(a) )`
- b. `b=[2.0] ; print( type(b) )`
- c. `c=(2) ; print( type(c) )`

ඉහත කේත ජේළි ධාවනයෙන් පසු ලැබෙන ප්‍රතිදාන පිළිවෙළින් දැක්වෙන්නේ,

- i) int , int , int    ii) float , float , int    iii) set , float , int    iv) set , list , int    v) set , list , tuple

130. පහත දැක්වෙන ප්‍රකාශ අතරින් වලංගු ප්‍රකාශය තෝරන්න.

- a. `A={1,2,3,4,6,2}`
- b. `B=(1,2,3,3,2,3)`
- c. `C={1:1,2:4,3:9,4:16}`
- d. `D=[]`

- i) දත්ත වර්ගය (variable Type) පිළිවෙළින් වන්නේ Dictionary , tuple , Dictionary , List වේ.
- ii) `A[2]` ලෙස ධාවනයෙන් 3 ප්‍රතිදානය වන අතර `C['2']` ධාවනයෙන් දෝෂ පණිවිඩ දර්ශනය වේ.
- iii) `D=D+[1,2,3,4,5]` ධාවනයෙන් පසු (run) කිසිම ප්‍රතිදානයක් ලබා නො දේ.
- iv) `C[1]=100` ලෙස ධාවනයේදී දෝෂ සමහර විට ඇති විය හැකියි.
- v) `C=A+B` සමහරක් විට දෝෂ පණිවිඩ ලබා දිය හැකි ය.

131. Tuple සඳහා එය නිර්මාණය කරන විට දී ඇතුළු කළ දත්ත වලට අමතරව තවත් දත්ත නැවත නැවත ඇතුළු කිරීමට නො හැකි ය. පහත ප්‍රකාශ සලකා ඒවායින් වලංගු (no errors) ප්‍රකාශ තෝරන්න.

```
A = 1, 2, 3, 4, 5, 6
```

- a. `print(A[1])`
- b. `A[1]=10`
- c. `A=A+(6,2,2)`

- i) a පමණි.    ii) b පමණි.    iii) a , b පමණි.    iv) a , c පමණි.    v) කිසිවක් නොවේ.

132. පහත කේතය (codes) සලකන්න. ඒ අනුව නිවැරදි වගන්තිය තෝරන්න.

```
>>>a=(1,2,3,4,5)
>>>a=a+(6,7)
>>>print(a)
(1,2,3,4,5,6,7)
```

(OUTPUT) කේත කොටස් ධාවනයෙන් පසු a හී ඇති අගය මෙය වේ.

- i) Tuple සඳහා නැවත දත්ත ඇතුළත් කළ නො හැකි බැවින් දෝෂ පණිවිඩ ලබා දේ.
- ii) Tuple Update කළ නො හැකි යන්න වැරදි ප්‍රකාශයක් බව මෙයින් පැහැදිලි වේ.

- iii) **Tuple** සඳහා අලුතින් දත්ත ඇතුළත් කළ නො හැකි ය. මෙහි දී සිදු කර ඇත්තේ a නමින් නැවත අලුත් **Tuple** එකක් සඳා එයට නැවත අගයන් එක් කිරීමයි.
- iv) **Python 3.78 update** එකත් සමග සඳහා නව දත්ත ඇතුළු කිරීමේ හැකියාව ලැබුණා.
- v) එය අහඹු සිදු වීමකි.

133. පහත කේත (codes) හි ප්‍රතිදානය (output) සලකන්න.

- a. `A = ( 6 ) ; print( type( A ) )`
- b. `B = ( 6 , ) ; print( type( B ) )`

- i) `int, tuple`      ii) `tuple, tuple`      iii) `int, int`      iv) `int, Error`      v) `Error, Error`

134. පහත (codes) සලකන්න. ඒ අතරින් වැරදි ප්‍රකාශ සලකන්න.

- a. `>>> A = ( 1 , 2 , 3 , 4 , 5 ) + ( 6 )`
- b. `>>> B = ( 1 , 2 , 3 , 4 , 5 ) + ( 6 , )`
- c. `>>> C = ( 1 , 2 , 3 , 4 , 5 ) + ( 6 , 7 )`

- i) a පමණි.      ii) b පමණි.      iii) a , b පමණි.      iv) a , c පමණි.      v) කිසිවක් නොවේ.

135. පහත කේත codes කොටස් සලකන්න.

```
A = ( ' a ' , ' b ' , ' c ' , ' d ' , ' 3 ' )
print ( A [ 0 ] )
print ( max( A ) )
print ( min ( A ) )
print ( len ( A ) )
print ( type( A [ 4 ] ) )
```

කේතය (codes) ධාවනයෙන් පසු ලැබෙන ප්‍රතිදානය (output) පිළිවලින් වන්නේ,

- i) `a , 3 , d , 5 , string`      ii) `' a ' , ' a ' , ' 3 ' , 5 , ' 3 '`      iii) `a , a , d , 4 , int`
- iv) `a , a , d , 5 , int`      v) `a , d , 3 , 5 , string`

136. පහත කේතය (codes) හි ප්‍රතිදානය (output) වන්නේ,

```
A = ( 1 + 3 , 3 + 4 , 6 , 6 )
print ( A * 2 )
```

- i) `( 4 , 7 , 6 , 6 )`      ii) `( 1 , 3 , 3 , 4 , 6 , 6 )`      iii) `( 4 , 7 , 6 , 6 , 4 , 7 , 6 , 6 )`
- iv) `error`      v) `( 4 , 4 , 7 , 7 , 6 , 6 , 6 , 6 )`

137. පහත කේතය (codes) හි ප්‍රතිදානය (output) වන්නේ,

```
Y = " YOYO "
T = tuple( Y )
print(T)
```

- i) `( ' YOYO ' )`      ii) `( ' Y ' , ' O ' , ' Y ' , ' O ' )`      iii) `( YOYO , )`      iv) `( Y , O , Y , O )`      v) `Error`

138. පහත කේතය (codes) හි ප්‍රතිදානය (output) වන්නේ,

```
>>> ( 1 , 2 , 3 ) + ( 10 , 20 , 30 )
```

- i) `( 1 , 2 , 3 , 0 , 0 , 0 )`      ii) `( 1 , 2 , 3 , ( 10 , 20 , 30 ) )`      iii) `( 1 , 2 , 3 , 10 , 20 , 30 )`
- iv) `( 1 , 2 , 3 , [ 10 , 20 , 30 ] )`      v) `Error`

139. පහත කේතය (codes) හි ප්‍රතිදානය (output) වන්නේ,

```
>>> (1, 2, 3) == (3, 2, 1)
```

- i) True      ii) true      iii) false      iv) False      v) O.K

140. පහත කේතය (codes) හි ප්‍රතිදානය (output) වන්නේ,

```
>>> (1, 2, 3, 4) + [6, 7, 8, 9]
```

- i) [1, 2, 3, 4, 5, 6, 7, 8, 9]      ii) (1, 2, 3, 4, 5, 6, 7, 8, 9)  
iii) {1, 2, 3, 4, 5, 6, 7, 8, 9}      iv) (1, 2, 3, 4, [5, 6, 7, 8, 9,])  
v) Error

141. පහත කේතය (codes) හි ප්‍රතිදානය (output) වන්නේ,

```
>>> (1, 2, (1+2), 4) * 2
```

- i) (1, 2, 3, 4, 1, 2, 3, 4)      ii) (1, 2, (1+2), 4, 1, 2, (1+2), 4)  
iii) (1, 1, 2, 2, 3, 3, 4, 4)      iv) (1, 2, (3), 4, 1, 2, (3))  
v) (2, 4, 6, 8)

142. පහත කේතය (codes) හි ප්‍රතිදානය (output) වන්නේ,

```
>>> [1, 2, 3, 4] + [6, 7, 8, 9]
```

- i) [1, 2, 3, 4, 5, 6, 7, 8, 9]      ii) (1, 2, 3, 4, 5, 6, 7, 8, 9)  
iii) {1, 2, 3, 4, 5, 6, 7, 8, 9}      iv) (1, 2, 3, 4, [5, 6, 7, 8, 9,])  
v) Error

143. පහත කේතය (codes) හි ප්‍රතිදානය (output) වන්නේ,

```
>>> X = (1, 2, 3, 4, 5) [::2]; print(X)
```

- i) (2, 4)      ii) (2)      iii) (3)      iv) (1, 3, 5)      v) Error

144. පහත කේතය (codes) හි ප්‍රතිදානය (output) වන්නේ,

```
>>> X = ('a', 'b', 'c', 'd', 'e', 'f') [1:5:2]; print(X)
```

- i) ('b', 'f', 'c')      ii) ('b', 'd', 'f')      iii) ('b', 'd')      iv) b, d      v) 'b', 'f', 'c'

145. පහත කේතය (codes) හි ප්‍රතිදානය (output) වන්නේ,

```
>>> X = (1, 2, 3) [1]; print(X)
```

- i) 2      ii) '2'      iii) [1]      iv) (2)      v) 1

146. පහත කේතය (codes) කොටස් සලකන්න.

```
A = ('a', 'b', 'c', 'd', '3')
```

```
print(A[1])
```

```
print(A[-1])
```

```
print(A[-2])
```

කේතය (codes) ධාවනයෙන් පසු ලැබෙන ප්‍රතිදානය (output) පිළිවෙලින් වන්නේ,

- i) 'b', '3', 'd'      ii) 'b', 'a', 'c'      iii) "a", "3", "d"  
iv) 'a', '3', 'd'      v) b, 3, d

147. පහත කේතය (codes) හි ප්‍රතිදානය (output) වන්නේ,

```
>>> X = (1, 2, 3) [1]; print(X)
```

- i) 2      ii) '2'      iii) [1]      iv) (2)      v) 1

148. පහත කේතය (codes) හි ප්‍රතිදානය (output) වන්නේ,

```
C = ('a', 'y', 't', 'c', 'p', 'o')
C.reverse()
print(a1)
```

- i) ('a', 'y', 't', 'c', 'p', 'o')
- ii) ('o', 'p', 't', 'c', 'y', 'a')
- iii) ('o', 'p', 'c', 't', 'y', 'a')
- iv) ('a', 'c', 'o', 'p', 't', 'y')
- v) Error

149. පහත කේතය (codes) හි ප්‍රතිදානය (output) වන්නේ,

```
C = ('a', 'v', 'x', 's', 'u', 'l')
C.sort()
print(a1)
```

- i) ('a', 'y', 't', 'c', 'p', 'o')
- ii) ('o', 'p', 't', 'c', 'y', 'a')
- iii) ('o', 'p', 'c', 't', 'y', 'a')
- iv) ('a', 'c', 'o', 'p', 't', 'y')
- v) Error

150. Tuple එකක් සම්බන්ධයෙන් පහත ප්‍රකාශ සලකන්න. ඒවා අතරින් වැරදි ප්‍රකාශ තෝරන්න.

- a. Tuple යනු mutable variable එකකි.
- b. Tuple සඳහා නැවත නැවත දත්ත ඇතුළු කළ නොහැකි විය හැක.
- c. Tuple නිර්මාණයේ දී බහුලව “ () ” භාවිත වන අතර “ [] ” භාවිතය අඩු වෙමින් පවතී.
- d. ( a ) = 1 , 2 , 3 , 4 යනු Tuple එකක් නිර්මාණය කරන නිවැරදි ආකාරයකි.

- i) c , d පමණි.
- ii) b , c පමණි.
- iii) b , d පමණි.
- iv) a , c පමණි.
- v) d පමණි.

>>> පහත කේත (code) කොටස සලකා 151, 152 ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
D = ('a', 'b', 's', 'e', 'c')
D.sort(); D.reverse()
print(D)
```

151. ඉහත කේතයේ ප්‍රතිදානය (output) දෝෂ පණිවිඩයකි. මෙයට හේතුව වන්නේ,

- i) Tuple සඳහා මෙම functions (sort , reverse) සහය නොදැක්වීම( Unsupported ).
- ii) Sort function එක ඉලක්කම් සඳහා පමණක් යොදා ගනී.
- iii) Tuple සඳහා sort function එක පමණක් සහය නොදැක්වීම( Unsupported ).
- iv) Tuple සඳහා reverse function එක පමණක් සහය නොදැක්වීම( Unsupported ).
- v) ඉහත කිසිවක් නොවේ.

152. ඉහත 151 දෝෂය (Error) නිවැරදි කිරීමට ගත හැකි ක්‍රියාමාර්ගය මොනවා ද? ( ඉහත කේතයේ කොටස් වෙනසකට භාජනය නොවේ. අලුතින් කේත පේළි( code lines ) එකතු වීමක් පමණක් සිදු වේ. )

- i) ඉහත කේතයේ අවසන් පේළියේ (Line) D = [ 'a', 'b', 's', 'e', 'c' ] ලෙස යෙදීම.
- ii) ඉහත කේතයේ print( D ) පෙර පේළියේ D . list( ) ලෙස යෙදීම.
- iii) D = ('a', 'b', 's', 'e', 'c') ට පසු පේළියේ C = D . list( ) ලෙස යෙදීම.
- iv) D = ('a', 'b', 's', 'e', 'c') ට පසු පේළියේ D = list( D ) ලෙස යෙදීම.
- v) කේතයේ පළමු පේළියේ D = [ 'a', 'b', 's', 'e', 'c' ] ලෙස යෙදීම.

153. පහත කේතය සලකන්න. එහි ප්‍රතිදානය කරවන ප්‍රතිදාන දෙපළ වන්නේ,

```
a = 'YOYO'
print(a[2])
b = list(a)
print(b)
```

- i) 'Y', ['Y', 'O', 'Y', 'O']
- ii) Y, ['Y', 'O', 'Y', 'O']
- iii) [Y], ['YOYO']
- iv) Error, ['Y', 'O']
- v) Y, ['Y', 'O']

>>> පහත කේත (code) කොටස සලකා 154, 155 ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
Y = 'YOYO'
Y[2]
print(len(Y)) → x
print(Y) → y
```

154. x ප්‍රතිදානය වන්නේ,  
 i) 1                      ii) 2                      iii) 3                      iv) 4                      v) Error → this is not a tuple or list
155. y ප්‍රතිදානය වන්නේ,  
 i) a                      ii) 'YaYO'                      iii) 'YOaY'                      iv) YOYO                      v) Error

>>> පහත දක්වා ඇති කේත (code) කොටස සලකා 156, 157 ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
A = 'YOYO productions'
B = len(A); print(B)
C = 'The Best '
D = C + A; print(D)
```

156. ඉහත code එකෙන් ලැබෙන පළමු ප්‍රතිදානය වන්නේ,  
 i) 14                      ii) 15                      iii) 16                      iv) 17                      v) 19
157. ඉහත code එකෙන් ලැබෙන දෙවන ප්‍රතිදානය වන්නේ,  
 i) The Best YOYO productions.  
 ii) The Best YOYO Productions  
 iii) 'The Best YOYO productions'  
 iv) "The Best YOYO productions"  
 v) Error – Can't Add String Value like Integer Value

>>> පහත දක්වා ඇති කේතය (code) කොටස සලකා 158 සිට 165 දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
S = "Bimsara S"
print(S[0] + S[-1]) ← 1
print(S[1:6]) ← 2
print(S[1::2]) ← 3
print(S[0:3:3]) ← 4
print(S[::2]) ← 5
print(S[:]) ← 6
print(S[0:-1:2]) ← 7
print(len(S)) ← 8
```

158. ඉහත කේතයේ (Code) 1 හි ප්‍රතිදානය (output) වන්නේ,  
 i) Error                      ii) 'Bi'                      iii) Bimsara S                      iv) BS                      v) S
159. ඉහත කේතයේ (Code) 2 හි ප්‍රතිදානය (output) වන්නේ,  
 i) Bimsara                      ii) imsar                      iii) B : i                      iv) rasmiB                      v) Bimsar
160. ඉහත කේතයේ (Code) 3 හි ප්‍රතිදානය (output) වන්නේ,  
 i) isr                      ii) Bmaa                      iii) Error                      iv) iBm                      v) Nothing
161. ඉහත කේතයේ (Code) 4 හි ප්‍රතිදානය (output) වන්නේ,  
 i) m                      ii) Bm                      iii) Error                      iv) Bsa                      v) B

162. ඉහත කේතයේ ( Code ) 5 හී ප්‍රතිදානය ( output ) වන්නේ,

- i) BmaaS                      ii) BmaS                      iii) Error                      iv) 'BmaaS'                      v) BBm

163. ඉහත කේතයේ ( Code ) 6 හී ප්‍රතිදානය ( output ) වන්නේ,

- i) Bimsara S                      ii) BmaS                      iii) Error                      iv) Nothing                      v) BBB

164. ඉහත කේතයේ ( Code ) 7 හී ප්‍රතිදානය ( output ) වන්නේ,

- i) BSm                      ii) Bmaa                      iii) Error                      iv) isr                      v) ' isr '

165. ඉහත කේතයේ ( Code ) 8 හී ප්‍රතිදානය ( output ) වන්නේ,

- i) 7                      ii) 8                      iii) 9                      iv) 10                      v) 11

166. පහත කේතය ( Code ) හී ප්‍රතිදානය වන්නේ,

```
A = [ 10 , 20 ,30 ,40 , 50 ]
A [ 4 ] = 50.0
Print ( A )
```

- i) [10,20,30,40,50,0,50]                      ii) [10,20,30,40,50,50]                      iii) [10,20,30,40,50]  
iv) [10,20,30,40,50.0]                      v) Error

167. පහත කේතය ( Code ) හී ප්‍රතිදානය වන්නේ,

```
A = [ 10 , 20 , 30 , 40 , 50 ]
A [ 5 ] = 60
print ( A )
```

- i) [10,20,30,40,50,0,50]                      ii) [10,20,30,40,50,50]                      iii) [10,20,30,40,50]  
iv) [10,20,30,40,50.0]                      v) Error

168. පහත කේතය සලකන්න.

```
A = [ ' a ' , ' b ' , ' c ' , ' d ' , ' e ' ]
A [ : ] = ' X '
print ( A )
```

- i) Error                      ii) X                      iii) [ ' X ' , ' a ' , ' b ' , ' c ' , ' d ' , ' e ' ]                      iv) [X]                      v) [ ' X ' ]

169. පහත කේතය ( Code ) හී ප්‍රතිදානය වන්නේ,

```
A = [ 10 , 20 , 30 , 40 , 50 ]
A [ 1:3 ] = 60
print ( A )
```

- i) [10,60,30,40,50]                      ii) [10,20,60,40,50]                      iii) [10,20,30,60,50]  
iv) [10,60,30,60,50]                      v) Error

170. පහත කේතය ( Code ) හී ප්‍රතිදානය වන්නේ,

```
A = [ 10 , 20 , 30 , 40 , 50 ]
A [ 1:3 ] = [60]
print ( A )
```

- i) [10,60,40,50]                      ii) [10,20,60,40,50]                      iii) [10,20,60,50]  
iv) [10,60,30,60,50]                      v) [10,60,30,40,60]

171. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
A = [ 10 , 20 , 30 , 40 , 50 ]
A [ 1:1 ] = [60]
print ( A )
```

- i) [10,20,30,40,50]                      ii) [10,60,20,30,40,50]                      iii) [10,60,20,50]  
iv) [60,10,20,30,60,50]                      v) Error

172. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
A = [ 10 , 20 , 30 , 40 , 50 ]
A [ 0 : 5 ] = [60]
print ( A )
```

- i) [10,60,30,40,50,60]                      ii) [10,20,30,40,50,60]                      iii) [60]  
iv) [10,60]                      v) Error

173. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
N = [ 9 , 8 , 7 , 6 , 5 ]
N [ 0 : -1 ] = [0]
print ( N )
```

- i) [9,8,7,6,5,0]                      ii) [0,5]                      iii) [5,0]                      iv) [9,8,7,6,0]                      v) [0]

174. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
N = [ 9 , 8 , 7 , 6 , 5 ]
N [ 0 : -3 ] = [10]
print ( N )
```

- i) [10,9,8,7,6,5]                      ii) [0,6,5]                      iii) [10,7,6,5]                      iv) [10,6,5]                      v) පිළිතුර නැත.

175. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
N = [ 1 , 2 , 3 , 4 , 5 ]
N [ -3 : -1 ] = [0]
print ( N )
```

- i) [1,2,3,0]                      ii) [1,2,0,5]                      iii) [1,2,0]                      iv) [0,3,4,5]                      v) [1,0,4,5]

176. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
N = [ a , b , c , d , c ]
N [ 0 : 4 ] = [0]
print ( N )
```

- i) [0, c]                      ii) [0, b, c]                      iii) [a, 0, b]                      iv) [a, 0, n]                      v) Error

177. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
N = [ 'a' , 'c' , 'b' , 'e' , 'g' ]
N [ 0 : 4 : 2 ] = [ 's' ] ; print ( N )
```

- i) [ 'e' , 'g' ]                      ii) [ 's' , 'c' , 'e' , 'g' ]                      iii) [ 's' , 'a' , 'b' , 'e' ]  
iv) [ ]                      v) Error



178. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
E = [ 1 , 2 , 3 , 1 , 2 , 3 ]
E . append ( ' w ' )
print ( E )
```

- i) [1,2,3,1,2,3,w]      ii) [ 1,2,3,1,2,3,'w' ]      iii) ['w',1,2,3,1,2,3]      iv) [1,2,3,1,2,3,'W']  
v) Error

179. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
p = [ 1 , 2 , 3 , 1 , 2 , 3 ]
p . insert ( ' o ' )
print ( p )
```

- i) [1,2,3,1,2,3,o]      ii) [ 1,2,3,1,2,3,'o' ]      iii) ['o',1,2,3,1,2,3]      iv) [1,2,3,1,2,3,'o']  
v) Error

180. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
p = [ 1 , 2 , 3 ] + [ 9 , 8 , 7 ]
print ( p )
```

- i) [1,2,3,9,8,7]      ii) [1,2,3,7,8,9]      iii) [10,10,10]      iv) [10]      v) [ [1,2,3] [9,8,7] ]

181. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
p = [ 1 , 2 , 3 ]
P = [ ' a ' , ' g ' , ' i ' ]
print ( p + P )
```

- i) ['a','g','i',1,2,3]      ii) [ [1,2,3] ['a','g','i'] ]      iii) [1,2,3,a,g,i]      iv) [1,2,3,'a','g','i']  
v) ඉහත කිසිවක් නොවේ.

182. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
p = [ 1 , 2 , 3 ]
q = [ 0 , 9 , 8 , 2 ]
p . extend ( q )
print ( p )
```

- i) [0,1,2,3,8,9]      ii) [1,2,3,0,9,8]      iii) [1,2,3,0,9,8,2]      iv) (1,2,3,0,9,8,2)      v) None

183. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
x = [ 8 , 9 , 6 , 5 ] ; y = [ 1 , 2 , 3 , 8 , 9 ]
y . extend ( x ) ; print ( x )
```

- i) [8,9,6,5]      ii) [1,2,3,5,6,8,9]      iii) [1,2,3,5,6,8,8,9,9]      iv) [8,9,6,5,1,2,3,8,9]      v) Error

184. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
g = [3,3,3]
g = g . extend ( g )
print ( g )
```

- i) [3,3,3,3,3,3]      ii) [3,3,3]      iii) None      iv) No      v) NONE

185. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
e = [ 1 , 2 , 3 ] + [ 7 , 8 , 9 ]
print ( e )
```

- i) [1,2,3,9,8,7]      ii) [1,2,3,7,8,9]      iii) [10,10,10]      iv) [10,]      v) Error

186. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
e = [1, 2, 3, 4, 5]
e . reverse ( )
print ( e )
```

- i) [1,2,3,4,5]      ii) [1,2,3,4,5,]      iii) [5,4,3,2,1]      iv) [5,]      v) Error

187. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
r = [ ' b ' , ' n ' , ' m ' ]
a = r . index ( ' b ' ) ; print(a)
```

- i) 0      ii) 1      iii) None      iv) [0]      v) Error

188. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
r = [ 1 , 2 , 1 , 3 ]
r . sort ( ) ; print ( r )
```

- i) [1,1,2,3]      ii) [1,2,3]      iii) None      iv) [1,2,3,]      v) Error

189. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
p = ( 1 , 2 , 3 )
p . sort ( ) ; print ( p )
```

- i) [1,2,3,]      ii) [3,2,1]      iii) [1,2,3]      iv) None      v) Error

190. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
p = "roo"
r = list ( p ) ; print(r)
```

- i) ['r','o','o']      ii) ['r','o']      iii) ['r','o','o']      iv) None      v) Error

191. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
p = 12
f = str ( p )
print ( f ) ; print ( type ( p ) )
```

- i) 12 , Int      ii) "12" , String      iii) '12' , String      iv) 12 , String      v) Error

192. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
k = " A "
l = int ( k ) ; print ( l )
```

- i) A      ii) 0      iii) 1 (first Alphabet)      iv) None      v) Error

193. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
t = [ 1 , 2 , 3 ]
str ( t ) ; print ( t )
```

- i) "1,2,3"      ii) ['1','2','3']      iii) "123"      iv) "[1,2,3]"      v) Error

194. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
t = [ 1 , 2 , 3 ]
print ( int ( t ) )
```

- i) 123      ii) 1,2,3      iii) [1,2,3]      iv) None      v) Error

195. පහත කේතය ( Code ) හි ප්‍රතිදානයන් වන්නේ,

```
t = "12" ; print ( int ( t ) )
```

- i) 12      ii) 1,2      iii) '12'      iv) (1,2)      v) Error

196. පහත කේතය ( Code ) හි ප්‍රතිදානයන් වන්නේ,

```
t = 12 > 10
print ( t )
```

- i) 'Yes'      ii) true      iii) True      iv) '12>10'      v) Error

197. පහත කේතය ( Code ) හි ප්‍රතිදානයන් වන්නේ,

```
t = 12 == 10
print ( t ) ; print ( type ( t ) )
```

- i) False , Int      ii) False, Boolean      iii) True, Bool      iv) 12 or 10      v) Error

198. පහත කේතය ( Code ) හි ප්‍රතිදානයන් වන්නේ,

```
F = [True, False, True]
print ( F ) ; print ( type ( F ) )
```

- i) [ True , False , True ] , Boolean      ii) [ 1 , 0 , 1 ] , List      iii) [ True , False , True ] , List  
iv) None      v) Error

199. පහත කේතය ( Code ) හි ප්‍රතිදානයන් වන්නේ,

```
F = [True, False, True]
print ( F ) ; print ( type ( F[1] ) )
```

- i) [ True , False , True ] , Boolean      ii) [ 1 , 0 , 1 ] , List      iii) [ True , False , True ] , List  
iv) None      v) Error

200. පහත කේතය ( Code ) හි ප්‍රතිදානයන් වන්නේ,

```
>>>F = True
>>>print ( int ( F ) )
>>>str( F )
```

- i) 1 , 1      ii) 1 , Error      iii) 1 , '1'      iv) 1 , 'True'      v) Error, Error

201. පහත කේතය ( Code ) හි ප්‍රතිදානය (OUTPUT) වන්නේ,

```
F = int( false )
F = bool ( F ) ; print ( F )
```

- i) true      ii) True      iii) 1      iv) None      v) Error

202. පහත කේතය ( Code ) හි ප්‍රතිදානය (OUTPUT) වන්නේ,

```
F = int ( 1 )
```

```
F = bool (F) ; print ( F )
```

- i) true                      ii) True                      iii) 1                      iv) None                      v) Error

203. පහත කේතය ( Code ) හි ප්‍රතිදානය (OUTPUT) වන්නේ,

```
F = int ( 0 )
```

```
F = bool (F) ; print ( F )
```

- i) false                      ii) False                      iii) 0                      iv) None                      v) Error

204. පහත කේතය ( Code ) හි ප්‍රතිදානය (OUTPUT) වන්නේ,

```
v = int ( 3 )
```

```
v = bool (v) ; print ( v )
```

- i) True                      ii) False                      iii) 0                      iv) 1                      v) Error

205. පහත කේතය ( Code ) හි ප්‍රතිදානය (OUTPUT) වන්නේ,

```
F = { 2 , 6 , 6 , 7 , 1 , 1 , 8 , 10 }
```

```
A = list(F) ; print(A)
```

- i) {1,1,2,6,6,7,8,10}                      ii) [1,2,6,7,8,10]                      iii) [2,6,6,7,1,1,8,10]  
iv) {1:2, 2:1, 6:2, 7:1, 8:1, 10:1}                      v) Error

206. පහත කේතය ( Code ) හි ප්‍රතිදානය (OUTPUT) වන්නේ,

```
F = { 2 , 6 , 6 , 7 , 1 , 1 , 9 , 10 }
```

```
F = print ( F )
```

- i) {1,1,2,6,6,7,8,10}                      ii) No outputs                      iii) {1,2,6,7,9,10}  
iv) {1:2, 2:1, 6:2, 7:1, 8:1, 10:1}                      v) Error

207. පහත කේතය ( Code ) හි ප්‍රතිදානය (OUTPUT) වන්නේ,

```
u = { 2 , 6 , 6 } == { 6 , 2 }
```

```
print ( u )
```

- i) {6,2}                      ii) 0                      iii) False                      iv) 1                      v) True

208. පහත කේතය ( Code ) හි ප්‍රතිදානය (OUTPUT) වන්නේ,

```
u = [ 2 , 6 , 6 ] == [ 6 , 2 ]
```

```
print ( u )
```

- i) {6,2}                      ii) 0                      iii) False                      iv) 1                      v) True

209. පහත කේතය ( Code ) හි ප්‍රතිදානය (OUTPUT) වන්නේ,

```
u = ( 2 , 6 , 6 ) == ( 2 , 6 , 6 )
```

```
print ( u )
```

- i) (6,2)                      ii) 0                      iii) False                      iv) 1                      v) True

210. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
o = { 'a', 'z', 'y' }
print ( o )
```

- i) {'a','z','y'}      ii) {'a','y','z'}      iii) {'z','y','a'}      iv) {'a','z','y'}      v) No special Order

211. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
d = " Bimsara "
d = set ( d ) ; print( d )
```

- i) {'a','m','r','i','s','B'}      ii) {'B','i','m','s','a','r','a'}      iii) {'a','r','a','s','m','i','b'}  
iv) {a,B,}      v) None

>>> පහත කේතය ( Code ) සලකා 212, 213, 214, 215 ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
>>> A = ( 25, 26, 13, 10 )
>>> S = [ 's', 'a', 'e', 'r', 'y' ]
>>> D = { 1, 3, 2, 1 }
```

212. >>> max ( A ) කේතය ( code ) ධාවනය කිරීමෙන් පසු ලැබෙන ප්‍රතිදානය වන්නේ,

- i) 10      ii) '26'      iii) 26      iv) 1      v) True

213. >>> max ( S ) කේතය ( code ) ධාවනය කිරීමෙන් පසු ලැබෙන ප්‍රතිදානය වන්නේ,

- i) 's'      ii) 'a'      iii) 'e'      iv) 'r'      v) 'y'

214. >>> max ( D ) කේතය ( code ) ධාවනය කිරීමෙන් පසු ලැබෙන ප්‍රතිදානය වන්නේ,

- i) 1      ii) '3'      iii) 3      iv) None      v) Error

215. >>> min ( D ) කේතය ( code ) ධාවනය කිරීමෙන් පසු ලැබෙන ප්‍රතිදානය වන්නේ,

- i) 1      ii) '3'      iii) 1, 1      iv) None      v) Error

216. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
u = [ 2, 6, 6 ]
u . reverse ( )
print ( u )
```

- i) [6, 6, 2]      ii) [6, 2]      iii) [2, 6, 6]      iv) [6, 2]      v) None

217. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
y = [ 'l', 't', 'q', 'e' ]
y . reverse ( )
print ( y )
```

- i) ['e','t','q','l']      ii) ['e','q','t','l']      iii) ['l','t','q','l']      iv) [ ]      v) None

218. පහත කේතය ( Code ) හි ප්‍රතිදානයන් වන්නේ,

```
y = [ 'L', 'N', 'M' ]
y . reverse ( ) ; y . sort ( )
print ( y )
```

- i) ['N','M','L']      ii) ['M','N','L']      iii) ['L','N','M']      iv) ['L','M','N']      v) None

219. පහත කේතය ( Code ) හි ප්‍රතිදානයන් වන්නේ,

```
E = ( 1 , 2 )
F = [ 10 , 20 , E ]
G = [ 100 , F ]
print ( G )
```

- i) { [ 10 , 20 , [ 1 , 3 ] ] }      ii) { 100 , ' F ' }      iii) [ 100 , [10, 20, (1, 2) ] ]  
iv) Error      v) { [ 10 , 20 , ( 1 , 3 ) , F , G ] }

>>> පහත කේතය ( Code ) සලකා ප්‍රශ්න 220 – 226 සඳහා පිළිතුරු සපයන්න. [පෙර ප්‍රශ්නයේ කේත පසු ප්‍රශ්නය සඳහා බල නොපායි. එනම් සෑම ප්‍රශ්නයකම පළමු පහත සඳහන් List එක නැවත නිකු ලෙසම පවතින බව සලකන්න]

```
L = [ 10 , 21 , 32 , 43 ]
```

220. L නම් වූ List එක පරිගණක මතකයෙන් සම්පූර්ණ වශයෙන් ම ඉවත් කිරීම අවශ්‍ය කේතය වන්නේ,

- i) L[] = ''      ii) L.del      iii) del L[ : ]      iv) del L      v) delete L

221. L නම් වූ List එක හි ඇති සම්පූර්ණ ම දත්ත පමණක් ඉවත් කිරීම අවශ්‍ය කේතය වන්නේ,

- i) L [ ] = ''      ii) L.del      iii) del L [ : ]      iv) del L      v) delete L

222. L නම් වූ List එක හි ඇති 21 දත්තය එහි ඉඩ සමගම ඉවත් කිරීම අවශ්‍ය කේතය වන්නේ,

- i) L [ 2 ] = ''      ii) L.del      iii) del L [ 21 ]      iv) del L      v) del L [ L . index (21) ]

223. >>> L [ 2 ] = '' ලෙස අදාළ දත්තය ඉවත් කිරීමට කේතයක් ( code ) ධාවනය කරන ලදී. ඉන් අනතුරුව >>> print ( len ( L ) ) කේතයක් ධාවනය කරන ලද විට OUTPUT එක වන්නේ,

- i) 3      ii) 4      iii) 5      iv) '3'      v) '4'

224. L නම් වූ List එක හි ඇති 21, 32 දත්ත එහි ඉඩ සමගම ඉවත් කිරීම අවශ්‍ය කේතය වන්නේ,

- i) L [ 1,3 ] = ''      ii) del L [ 1:4 ]      iii) del L [ 21 , 31 ]      iv) del L      v) del L [ 1:3 ]

225. L නම් වූ List එක හි ඇති 10, 32 ලෙස එකක් ඇර එකක් ඇති දත්ත එහි ඉඩ සමගම ඉවත් කිරීම අවශ්‍ය කේතය වන්නේ,

- i) L [ 0 : 3 ] = ''      ii) del L [ 0 : 2 ]      iii) del L [ : : 2 ]      iv) del L [ 2 ]      v) del L [ 0,3 ]

226. L [ 0 ] + L [ -1 ] මෙම කේතයේ ප්‍රතිදානය වන්නේ,

- i) -53      ii) 66      iii) 55      iv) 53      v) 45

227. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
u = ( 2 , 6 , 6 ) #Tuple
u . sort ( ) ; u . reverse ( )
print ( u )
```

- i) ( 6 , 6 , 2 )      ii) ( 6 , 2 )      iii) ( 2 , 6 , 6 )      iv) ( 6 , 2 )      v) Error

228. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
u = { 2 , 6 , 6 } #Set
u . sort ( ) ; u . reverse ( )
print ( u )
```

- i) { 6 , 6 , 2 }      ii) { 6 , 2 }      iii) { 2 , 6 , 6 }      iv) { 6 , 2 }      v) Error

229. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
r = {1,2,3,4,5}
a = r . index [2]
```

- i) 0                      ii) 1                      iii) None                      iv) [0]                      v) Error

230. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
S = ' YOYO Company 's Building '
```

- i) YOYO Company's Building                      ii) YOYO Company                      iii) s Buildings  
iv) Logical Error                      v) Syntax Error

>>> පහත කේතය ( Code ) සලකා ජරණ 231 – 234 සඳහා පිළිතුරු සපයන්න.

```
K = { 1 : "A" , 'B' : 1 , 3 : "C" , 4 : "B" }
```

231. >>> print ( K [ 1 ] ) ප්‍රතිදානය වන්නේ [OUTPUT] වන්නේ,

- i) A                      ii) 'A'                      iii) 1                      iv) B                      v) 'B'

232. >>> print ( K [ 'B' ] ) ප්‍රතිදානය වන්නේ [OUTPUT] වන්නේ,

- i) 1                      ii) '1'                      iii) 4                      iv) '4'                      v) Error

233. >>> print ( K [ 'C' ] ) ප්‍රතිදානය වන්නේ [OUTPUT] වන්නේ,

- i) 3                      ii) '3'                      iii) 4                      iv) '4'                      v) Error

234. K නම් Keys හා Value පිළිවලින් දැක්වෙන්නේ,

- i) 1,2,3,4 - 'A','B','C','D'                      ii) 'A','B','C','D' - 1,2,3,4                      iii) 'A',1,'C','D' - 1,'B',3,4  
iv) 1,'B',3,4 - 'A','1','C','D'                      v) 1,'B',3,4 - 'A',1,'C','D'

235. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
u = { 1 : ' A ' , 2 : ' B ' , 26 : ' Z ' }
R = u . keys ( )
print ( R )
```

- i) dict\_keys([1,2,26])                      ii) dict\_keys([A,B,Z])                      iii) { 1 : ' A ' , 2 : ' B ' , 26 : ' Z ' }  
iv) Error                      v) None

236. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
u = { 1 : ' A ' , 2 : ' B ' , 26 : ' Z ' }
R = u . values ( )
print ( R )
```

- i) dict\_values([1,2,26])                      ii) dict\_values(['A','B','Z'])                      iii) { 1 : ' A ' , 2 : ' B ' , 26 : ' Z ' }  
iv) Error                      v) None

237. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
hy = { 1 : [ 1 , 2 ] , 2 : ( ' d ' , ' f ' ) , 26 : { 1 , 3 } }
R = hy[1]
print ( R )
```

- i) ('d','f')                      ii) {1,2}                      iii) [1,2]                      iv) [1,2,3]                      v) Error or None

238. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
T, t = '92' ; print ( T )
```

- i) 9                      ii) 92                      iii) 29                      iv) 't'                      v) Error or None

239. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
T, t = 92 ; print ( T )
```

- i) 9                      ii) 92                      iii) 29                      iv) 't'                      v) Error

240. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
A = "Yoyo"
A . upper ( )
print ( A )
```

- i) YoYo                      ii) Yoyo                      iii) YOYO                      iv) yoyo                      v) 'YOYO'

241. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
A = "Yoyo"
B = A . upper ( )
print ( A )
```

- i) YoYo                      ii) Yoyo                      iii) YOYO                      iv) yoyo                      v) 'YOYO'

242. පහත කේතය ( Code ) හි ප්‍රතිදානය වන්නේ,

```
A = "Yoyo"
B = A . swapcase ( )
print ( B )
```

- i) YoYo                      ii) yOYO                      iii) YOYO                      iv) yoYO                      v) 'YoyO'

>>> පහත කේතය ( Code ) සලකා ප්‍රශ්න 243 – 244 සඳහා පිළිතුරු සපයන්න.

```
A = " Sltv "
T = A . upper ( ) ; print ( T ) ← 1
A = A . lower ( ) ; print ( A ) ← 2
B = A . swapcase ( ) ; print ( B ) ← 3
```

243. ඉහත කේතයේ ( Code ) 1 හි ප්‍රතිදානය ( output ) වන්නේ,

- i) sltv                      ii) 'SLTV'                      iii) SltV                      iv) sLTV                      v) SLTV

244. ඉහත කේතයේ ( Code ) 2 හි ප්‍රතිදානය ( output ) වන්නේ,

- i) sltv                      ii) 'sltv'                      iii) sLTV                      iv) sltV                      v) SLTV

245. ඉහත කේතයේ ( Code ) 3 හි ප්‍රතිදානය ( output ) වන්නේ,

- i) Sltv                      ii) sLTV                      iii) SLTV                      iv) sLtV                      v) slTv

246. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
S = " TOTO - YOYO "
A = S . replace ( " Y " , " T " )
print ( A )
```

- i) YOYO-TOTO                      ii) YOYO-YOYO                      iii) YYYY-YYYY                      iv) TOTO-TOTO                      v) TTTT-TTTT



247. ඉහත 246 හි සඳහන් කේතයේ ( Code ) පහළ ම ජේලියේ පහත කේතය යොදන ලදී. එවිට එහි ප්‍රතිදානය ( output ) වන්නේ,

```
print ( S )
```

- i) TOTO-YOYO      ii) YOYO-YOYO      iii) YYYY-YYYY      iv) TOTO-TOTO      v) TTTT-TTTT

248. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
S = "ABWSYZAZW"
```

```
A = S . replace ( max ( S ) , min ( S ) )
```

```
print ( A )
```

- i) Error or None      ii) 'AZAZAZAZAZAZAZAZ'      iii) 'ABWSYAAZW'      iv) 'ZBWSYZZZW'  
v) 'ABWSYAAAW'

249. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
S = "yoyo productions"
```

```
A = S . title ( )
```

```
print ( A )
```

- i) Yoyo productions      ii) - Yoyo productions -      iii) Yoyo\_Productions      iv) Error  
v) Yoyo Productions

250. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
S = "yOyO pRoductions"
```

```
A = S . title ( )
```

```
print ( A )
```

- i) Yoyo productions      ii) - Yoyo productions -      iii) Yoyo\_Productions      iv) Error  
v) Yoyo Productions

251. පහත කේතයේ (Code) හි ප්‍රතිදානය (output) වන්නේ,

```
S = "SL Tech Vision YouTube Channel"
```

```
A = 's' in S
```

```
print ( A )
```

- i) None      ii) sL Tech Vision YouTube Channel      iii) True      iv) False      v) S'L'

252. පහත කේතයේ (Code) හි ප්‍රතිදානය (output) වන්නේ,

```
A = "SL Tech Vision YouTube Channel"
```

```
A = 't' in A
```

```
print ( A )
```

- i) SL Tech Vision YouTube Channel      ii) 1      iii) True      iv) False      v) 0

253. පහත කේතයේ ( Code ) හි ප්‍රතිදානයන් පිළිවෙලින් ( output ) වන්නේ,

```
A = [1,4,5,8,0]
```

```
print ( sorted ( A ) )
```

```
print ( A )
```

- i) [0,1,4,5,8] – [0,1,4,5,8]      ii) [0,1,4,5,8] – [1,4,5,8,0]      iii) False – [1,4,5,8,0]  
iv) True – [1,4,5,8,0]      v) [8,5,4,1,0] – [1,4,5,8,0]

254. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
A = "SL Tech Vision YouTube Channel"
A = 'SL Tech' in A
print ( A )
```

- i) SL Tech Vision YouTube Channel      ii) 1      iii) True      iv) False      v) 0

255. පහත කේතයේ ( Code ) හි ප්‍රතිදානයන් පිළිවෙලින් ( output ) වන්නේ,

```
A = ( X for X in range ( 5 ) )
print ( A ) ; print ( type ( A ) )
```

- i) [1,2,3,4,5] - Tuple      ii) (0,1,2,3,4) - Tuple      iii) {1:1,2:2,3:3,4:4,5:5} - Dic  
iv) (1,2,3,4) - loop      v) Error – Can't add Values to Tuple

256. පහත කේතයේ ( Code ) හි ප්‍රතිදානයන් පිළිවෙලින් ( output ) වන්නේ,

```
C = { X for X in range ( 5 ) }
print ( C ) ; print ( type ( C ) )
```

- i) [1,2,3,4,5] - List      ii) {0,1,2,3,4} - Set      iii) {1:1,2:2,3:3,4:4,5:5} - Dic  
iv) {1,2,3,4} - Set      v) Error – Can't add Values to Set

257. පහත කේතයේ ( Code ) හි ප්‍රතිදානයන් පිළිවෙලින් ( output ) වන්නේ,

```
C = [ X for X in range ( 2 , 5 ) ]
print ( C )
```

- i) [2,5]      ii) [2,3,4,5]      iii) [2,3,4]      iv) [3,4,5]      v) [4,5]

258. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
C = [ X for X in range ( -1 , 5 , 2 ) ]
print ( C )
```

- i) [0,2,4]      ii) [-1,5,2]      iii) [-1,1,3,5]      iv) [-1,1,3]      v) [1,2,3,5]

259. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
C = [ X for X in reversed ( range ( -1 , 5 , 2 ) ) ]
print ( C )
```

- i) [0,1,4]      ii) [3,1,-1]      iii) [-1,1,3,5]      iv) [-1,1,3]      v) [4,2,0,-2]

260. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
C = [ X * 2 for X in reversed ( range ( -1 , 5 , 2 ) ) ]
print ( C )
```

- i) [6,2,-2]      ii) [-2,10,4]      iii) [-1,1,3,5]      iv) [-2,2,6]      v) Error

261. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
C = [ X for X in reversed ( range ( -1 , 5 , 2 ) ) ]
print ( C )
```

- i) [0,2,4]      ii) [-1,5,2]      iii) [-1,1,3,5]      iv) [3,1,-1]      v) [1,2,3,5]

262. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
C = { X : X ** 2 for X in reversed ( range ( 1 , 5 , 2 ) ) }
print ( C )
```

- i) {1:'1',3:'9'}                      ii) {3,9,1,1}                      iii) {3:9:-1:1}                      iv) {1,9}                      v) {1:1,3:9}

263. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
C = { 1 : ' A ' , 1 : ' B ' , 3 : ' C ' , 4 : ' D ' }
print ( C [ 1 ] )
```

- i) A                      ii) B                      iii) ['A','B']                      iv) AB                      v) Error

264. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
Z = 2.5e3
print ( Z )
```

- i) 2.50003                      ii) 7.5                      iii) 25000.0                      iv) 2500.0                      v) Error

265. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
Z = "axetyreyuieop"
A = Z . split ( 'e' )
print ( A )
```

- i) ('ax','tyr','yui','op')                      ii) ['ax','tyr','yui','op']                      iii) ['axe','tyre','yuie','ope']  
iv) axe,tyre,yuie,op                      v) 'axtyryuiop'

266. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
Z = "dog,cat,cow,lion"
A = Z . split ( ',' )
print ( A )
```

- i) (dog,cat,cow,lion)                      ii) [dog cat cow lion]                      iii) [dog,cat,cow,lion]  
iv) dog,cat,cow,lion                      v) 'dog,cat,cow,lion'

267. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
Z = "axetyreyuieop"
A = Z . split ( 'et' )
print ( Z )
```

- i) ('ax','tyr','yui','op')                      ii) ['ax','tyr','yui','op']                      iii) ['ax', 'reyuieop']  
iv) ax etyr eyui eop                      v) 'ax, tyryuiop'

268. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
Z = "axetyreyuieop"
A = Z . split ( )
print ( A )
```

- i) ('ax','tyr','yui','op')                      ii) ['ax','tyr','yui','op']                      iii) ['axe','tyre','yuie','ope']  
iv) ['axetyreyuieop']                      v) 'axetyreyuieop'

>>> පහත කේතය ( Code ) සලකා ප්‍රශ්න 269 – 271 සඳහා පිළිතුරු සපයන්න.

```
F = "++++YOYO++++"
G = "++++YOYO++++"
H = "++++YOYO++++"
F = F . strip ( " + " ) ; print ( F ) ← 1
G = G . lstrip ( " + " ) ; print ( G ) ← 2
H = H .rstrip ( " + " ) ; print ( H ) ← 3
```

269. ඉහත කේතයේ ( Code ) 1 හි ප්‍රතිදානය ( output ) වන්නේ,

- i) YOYO                  ii) +YOYO+                  iii) YOYO+                  iv) 'YOYO'                  v) '+YOYO+'

270. ඉහත කේතයේ ( Code ) 2 හි ප්‍රතිදානය ( output ) වන්නේ,

- i) YOYO++++                  ii) +YOYO                  iii) YOYO+                  iv) 'YOYO'                  v) ++++YOYO

271. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
A = "Y,O,Y,O"
V = A . split ( "," )
print ( V )
```

- i) ('Y','O','Y','O')                  ii) ['O','Y','O','Y']                  iii) ['Y','O','Y','O']                  iv) 'Y','O','Y','O'                  v) 'YOYO'

272. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
world = "Python"
d = 'Hello' + " world" ; print ( d )
```

- i) 'Hello world'                  ii) Hello world                  iii) Hello Python  
iv) 'Hello Python'                  v) Syntax Error

273. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
world = "Python"
d = 'Hello' + world ; print ( d )
```

- i) 'Hello world'                  ii) Hello world                  iii) HelloPython  
iv) 'Hello Python'                  v) Syntax Error

274. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
G = "yoyo prod"
D = "pro"
L = G in D ; print ( L ) ; print(D in G)
```

- i) Pro                  ii) Yoyo Prod                  iii) False True                  iv) True False                  v) Syntax Error

275. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
G = "SL Tech Vision"
H = "YouTube Channel"
Word = G [ 0 :: 2 ] + H [ 3 : 9 : 2 ] + 'S' ; print(Word)
```

- i) S ehVsoTbS                  ii) SehVsoTbS                  iii) S ehVsoTb S                  iv) SLYouChanIS                  v) S ehViin S

276. පහත කේතයේ ( Code ) හි ප්‍රතිදානය ( output ) වන්නේ,

```
G = "SL Tech Vision"
H = "YouTube Channel"
Word = G [ 0 : : 2 ] + H [ 3 : 9 : 2 ] + 'S'
print ( Word )
```

- i) S ehVsoTbS      ii) S ehVsoTb S      iii) SehVsoTbS      iv) SLYouChanIS      v) S ehViin S

277. පහත කේතයේ ප්‍රතිදානය ( output ) වන්නේ,

```
print(" I am % i years old. He is % i years old "% ( 9 , 10 ) )
```

- i) I am 9 years old.He is 10 years old.      ii) I am 10 years old.He is 9 years old.  
iii) I am 9 years old.He is 10.0 years old.      iv) I am 9.0000 years old.He is 10.0000 years old.  
v) ඉහත සියල්ලම නොවේ.

278. පහත කේතයේ ප්‍රතිදානය ( output ) වන්නේ,

```
print(" I am % f years old. He is % f years old "% ( 9 , 10 ) )
```

- i) I am 9 years old.He is 10 years old.      ii) I am 10 years old.He is 9 years old.  
iii) I am 9 years old.He is 10.0 years old.      iv) I am 9.0000 years old.He is 10.0000 years old.  
v) ඉහත සියල්ලම නොවේ.

279. පහත කේතයේ ප්‍රතිදානය ( output ) වන්නේ,

```
print (" I am % f years old. He is % i years old "% ( 9 , 10 ) )
```

- i) I am 9 years old.He is 10 years old.      ii) I am 10 years old.He is 9 years old.  
iii) I am 9.00000 years old.He is 10 years old.      iv) I am 9.00 years old.He is 10.00 years old.  
v) ඉහත සියල්ලම නොවේ.

280. ඉහත 278, 279 කේතයේ f හා i (Bolded Letters) වල අර්ථය සඳහන් වන ප්‍රකාශය වන්නේ,

- i) f මඟින් float වර්ගයේ හා i මඟින් int වර්ගයේ අගයන් ප්‍රතිදානය කරවන බව හැඟවීමට.  
ii) f මඟින් first ලෙස අර්ථ ගන්වා පළමුව වරහන් තුළ ඇති අගය එතනට ආදේශ කරයි.  
iii) i , f ට වඩා විශාල නිසා වරහන් තුළ ඇති අගයන් විශාලත්වයේ පිළිවෙලට දර්ශනය කරවයි.  
iv) සාමාන්‍ය ලෙස යෙදූ Variable Name එකකි.  
v) ඉහත සියල්ලම වැරදි.

281. පහත කේතයේ ක්‍රියාත්මක කිරීමේදී දෝෂ පණිවිඩයක් දීස් විය. එය නිවැරදි කරන්නේ,

```
print (" I am % s years old. He is % i years old "% ( 9 , '10' ) )
```

- i) 9ය '9' ලෙස වෙනස් කිරීම.      ii) දෝෂ රහිතයි.      iii) '10'ය 10 ලෙස කොමා ඉවත් කිරීම මගින්.  
iv) % i ය % s ලෙස වෙනස් කිරීම.      v) % s ය % i ලෙස වෙනස් කිරීම.

282. පහත කේතයේ ප්‍රතිදානය ( output ) වන්නේ,

```
print (" I am % f years old. He is % s years old "% ( 9 , 'Ten' ) )
```

- i) I am 9 years old.He is 10 years old.                      ii) I am 10 years old.He is 9 years old.  
 iii) I am 9.00000 years old.He is Ten years old.          iv) I am 9 years old.He is Ten years old.  
 v) ඉහත සියල්ලම නොවේ.

>>>. පහත කේතය සලකා 283 – 286 ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
a = 1
while a != False :
    a = int (input ( ) )
    if a == 1 :
        continue
    print("ideaPack")
```

283. ඉහත කේතයේ 1 ආදානය (input) කල විට. සිදුවන්නේ,

- i) ideaPack ලෙස print වන අතර තවත් ආදාන ලබා ගැනීමට සූදානම් වේ  
 ii) ideaPack ලෙස print වන අතර තවත් Program එක අවසන් වේ.  
 iii) ideaPack ලෙස print නොවන අතර තවත් ආදාන ලබා ගැනීමට සූදානම් වේ  
 iv) උඩුකොමා සහිතව ආදාන ලබා දුන් නිසා Error පණිවිඩ ලබා දේ.  
 v) සිදුවන දේ ස්ථිර වශයෙන් සඳහන් කිරීමට ඇති continue හැකියාව නිසා මග හැරී ඇත.

284. 2 , 1 , 6 , 99 , 1 , -1 ලෙස පිළිවෙලින් ආදානය කළ විට ideaPack ලෙස දිස් කරවන වාර ගණන වන්නේ,

- i) 4                      ii) 5                      iii) 3                      iv) 1                      v) Error

285. ඉහත continue වෙනුවට break ලෙස වෙනස් කළ විට ideaPack ලෙස දැක්වෙන වාර ගණන,

- i) 4                      ii) 5                      iii) 3                      iv) 1                      v) Error

286. ඉහත කේතයේ !False වෙනුවට !True ලෙස වෙනස් කරන ලදී. ඉන් පසු නිමල් විසින් 1 ආදානය කරන ලදී. එවිට ප්‍රතිදාන වන්නේ,

- i) ආදාන ලබා ගැනීමට පෙර කේතය අවසන්වේ.  
 ii) ප්‍රතිදාන ලබා නොදෙන අතර තව දුරටත් ක්‍රියාත්මක වේ.  
 iii) ලෙස එක් වරක් හෝ නොවේ.  
 iv) එක් වරක් පමණක් ලෙස ප්‍රතිදානය කරවයි.  
 v) ලෙස එක් වරක් වත් ප්‍රතිදානය නොවන අතර ඉහත ප්‍රශ්නයේ සඳහන් ප්‍රකාශය සමග එකඟ විය නොහැක.

>>> . පහත පයිතන් කේතය(python Code) කේතය සලකන්න. එමගින් 287 – 289 දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
a = input("Input Value : ")
b = int (a)
c = int ( b ** 0.5 )
print(c)
```

287. ඉහත කේතය සඳහා **100 Input** කල විට **Output** ලැබෙන එක වන්නේ,

- i) 1                      ii) 10.00                      iii) 10000                      iv) 10                      v) Error

288. ඉහත කේතය සඳහා **30 Input** කල විට **Output** ලැබෙන එක වන්නේ,

- i) 5.477225575051661                      ii) 5                      iii) 9000                      iv) 5.5                      v) Error

289. ඉහත කේතය සඳහා පිළිවෙලින් **1** හා **2** ආදානය කළ විට ලැබෙන **Output** පිළිවෙලින් වන්නේ,

- i) 1, 1                      ii) 1, 1.43                      iii) 0, 1                      iv) 1, Error                      v) Both Error

>>> . පහත පයිතන් කේතය (**python Code**) කේතය සලකන්න. **1** සිට **10** දක්වා ඉරට්ටේ අගයන් වල එකතුවට ආදානය කරන ලද අගය එකතු කළ විට එන අගය ගෙන, **100** එම අගයෙන් බෙදූ විට ලැබෙන අගය **Output** කිරීමට කේතය සකස් කර ඇත. එමගින් **290 – 293** දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
sum = int ( input( ) )
for count in range ( 1 , 11 , 2 ) :
    if count % 2 == 0 :
        sum = sum + count
value = int(100 / sum)
print(value)
```

290. ඉහත කේතය ක්‍රියාත්මක කිරීමේදී ලැබෙන ප්‍රතිදානය / **output** වන්නේ (**input = 1**),

- i) 3.333                      ii) 100                      iii) 5                      iv) 20                      v) Error

291. ඉහත කේතයේ සමහර **input** වලදී **Error** පණිවිඩ ලබා දේ, එයට හේතුවන කොටස සහිත ඡේලිය වන්නේ පහත ඒවා අතරින්,

- i) 1 ඡේලිය                      ii) 2 ඡේලිය                      iii) 3 ඡේලිය                      iv) 6 ඡේලිය                      v) අවසන් ඡේලිය

292. ඉහත කේතයේ **Error** පණිවිඩ ලබා දෙන **input** අගයක් වන්නේ,

- i) 1                      ii) 2                      iii) 3                      iv) 15                      v) 0

293. ඉහත කේතයේ පළමු ඡේලිය **sum = 0** ලෙස **edit** කරන ලදී. එතැන් සිට කේතය ධාවනයේදී සෑම විටම **Errors** දක්නට ලැබීණි. මෙයට හේතුව වන්නේ,

- i) **value = int (100 / sum)** සහිත කොටසේ **int** ලෙස සඳහන් කොටස.  
 ii) **For Loop** එක මගින් ලබාදෙන අගයන් එකතුව නිසා.  
 iii) **print(value)** හි **value variable** එක **String type variable** එකක් බවට හරවා නොතිබීම.  
 iv) **ZeroDivisionError: division by zero**  
 v) **Errors** පණිවිඩ ලබා දීමක් සිදු නොවේ කිසිවිටකත්.

294. පහත කේතය සලකන්න. එයට **4 INPUT** කරවීම එහි **OUTPUT** එක වන්නේ,

```
A = int (input ("Enter Triangle Height : "))
for p in range (1, A) :
    print ("#" * p )
```

- i) ###  
##  
#
- ii) ####  
###  
##  
#
- iii) #  
##  
###
- iv) #  
##  
###
- v) Error

295. පහත කේතය සලකන්න.

```
Total = 0
for k in range (10) :
    S = input ("Enter Values : ") ; S = int (S)
    Total = Total + S
print (Total)
```

ඉහත කේතය සඳහා පිළිවලින් 2, 4, 9, 10, 6, 1, 2, 8, 9, 10, 11, 3 input කරන ලදී. එවිට output වන අගය වන්නේ,

- i) 61                      ii) 74                      iii) 60                      iv) 51                      v) Error

296. පහත කේතය සලකන්න. එමගින් එම කේතයේ ප්‍රතිදානය / output විය හැක්කේ,

```
import random
A = random . randint (0 , 5)
print(A)
```

- i) 0 න් 5 න් අතර (5 රහිත) ඕනෑම අගයක්.  
ii) 0, 1, 2, 3, 4, 5 යන අගයන් වලින් එක අගයක්.  
iii) 0 න් 100000 න් අතර එක් අගයක්.  
iv) 1 න් 4 න් අතර ඇති ඕනෑම එක් අගයක්.  
v) 0 හෝ 5 යන ඉලක්කම් දෙකෙන් එකක්.

297. පහත කේතය සලකන්න. එමගින් එම කේතයේ ප්‍රතිදානය / output විය නොහැක්කේ,

```
import random
Total = 0;
for i in range (4) :
    A = random . randint (0 , 5)
    Total = Total + A
print(Total)
```

- i) 20                      ii) 26                      iii) 4                      iv) 17                      v) 6

298. පහත කේතය සලකා කේතය සම්බන්ධයෙන් පහත දක්වා ඇති ප්‍රකාශ අතරින් නිවැරදි වන්නේ,

```
while not True :
    print ("ideaPack YouTube Channel")
print ("ideaPack YouTube Channel")
```

- i) ideaPack YouTube Channel ලෙස අවසානයක් නොමැතිව Print වේ.  
ii) ideaPack YouTube Channel ලෙස එක් වරක්වත් Print නොවේ.  
iii) ideaPack YouTube Channel ලෙස එක් වරක් Print වේ.  
iv) ideaPack YouTube Channel ලෙස 2 වරක් Print වේ  
v) Displaying Error Messagers



299. පහත කේතය සලකන්න. එමගින් එම කේතයේ ප්‍රතිදානය / output විය හැක්කේ,

```
for c in range (0, 5, 2) :
    print( c , end = ',' )
```

- i) 0, 2, 4,      ii) 1, 3      iii) 1, 3, 5      iv) 2, 3, 4, 5      v) 1, 3, 5, 7

300. පහත කේතය සලකන්න. එමගින් එම කේතයේ ප්‍රතිදානය / output විය හැක්කේ,

```
for g in range (5, 0, -1) :
    print( g , end = ',' )
```

- i) 5,4,3,2,1      ii) 5,4,3,2,1,      iii) 4,3,2,1,      iv) 4,3,2,1      v) 1,2,3,4,5,

301. පහත කේතය සලකන්න. එමගින් එම කේතයේ ප්‍රතිදානය / output විය හැක්කේ,

```
for t in (1, 3, 5, 7, 9):
    print(t, end = ',')
```

- i) 1,3,5,7,9      ii) 1,2,3,4,5,      iii) 2,4,6,8,10      iv) 9,7,5,3,1      v) 1,3,5,7,9,

302. පහත කේතය සලකන්න. එමගින් එම කේතයේ ප්‍රතිදානය / output විය හැක්කේ,

```
for k in reversed (range ( 0, 5 ) ) :
    print (k , end = ',')
```

- i) 5,4,3,2,1,0      ii) 5,4,3,2,1      iii) 4,3,2,1,0,      iv) 0,1,2,3,4,5,      v) 1,2,3,4,5,

>>> පහත පයිතන් කේතය (python Code) කේතය සලකන්න. එමගින් 303 – 308 දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
count = 0
Value = int (input ("Enter Number in 1 to 10: "))
if ( Value >= 0 and Value <= 10 ) : ← # second line
    for m in range (1, 11) :
        if Value % m == 0 :
            count = count + 1
    if count == 2 :
        print(" 'ideaPack' ")
```

303. දෙවෙනි පේළියේ / 2<sup>nd</sup> Line ඇති කේතයේ අවශ්‍යතාවය වන්නේ,

- i) ආදානය / කරන අගය ඉලක්කමක්ද යන වග පරීක්ෂා කිරීම.  
 ii) ආදානය කරන අගය 1 – 10 දක්වා අගයක්ද යන වග පරීක්ෂාව  
 iii) ආදානය කරන අගය 0 – 10 අතර අගයක් නොවේ නම් පහළ කොටසට ගමන් කරයි.  
 iv) දත්ත පරාස පරීක්ෂාව  
 v) ඉහත කිසිවක් නොවේ.

304. 5 ආදානය / input කළ විට ලැබෙන ප්‍රතිදානය / output වන්නේ,

- i) 'ideaPack'      ii) ideaPack      iii) no output      iv) idea      v) Error

305. 4 ආදානය / input කළ විට ලැබෙන ප්‍රතිදානය / output වන්නේ,

- i) ideaPack      ii) 'ideaPack'      iii) no output      iv) idea      v) Error

**306. 10** ආදානය / **input** කළ විට ලැබෙන ප්‍රතිදානය / **output** වන්නේ,

- i) **ideaPack**            ii) **'ideaPack'**            iii) **no output**            iv) **idea**            v) **Error**

**307. ideaPack** ලෙස ප්‍රතිදානය / **output** කිරීමට සමත් පහත ආදානයන් / **input** වන්නේ,

- i) **2, 4, 3, 5, 11**            ii) **1, 4, 6, 8, 10**            iii) **2, 4, 8**            iv) **1, 2, 3, 5, 7**            v) **2, 3, 5, 7**

**308. ideaPack** ලෙස ප්‍රතිදානය / **output** කිරීමට සමත් වන අගයන් හඳුන්වන ගණිතමයම නාමය වන්නේ,

- i) බෙදෙන අගයන්            ii) වර්ග අගයන්            iii) හතරැස් අගයන්            iv) ප්‍රථමක අගයන්  
v) විශේෂිත නම් නැත

>>> පහත පයිතන් කේතය (**python Code**) කේතය සලකන්න. එමගින් **309 – 314** දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
a = b = c = -1
k = [0, 0, 0, 0, 0, 0, 0, 0]
val = int (input ("Enter a Value You want (0 – 255) : "))
while val != 1:
    a = val // 2
    b = val % 2
    val = a
    k[c] = b
    c = c - 1
k[c] = val
print (k)
```

**309. k** ලෙස ඇති **Variable** එක කුමන **Variable** වර්ගයේ එකක් ද,

- i) **integer**            ii) **String**            iii) **Tuple**            iv) **List**            v) **Dictionary**

**310. 10** අගය ආදානය / **input** කළ විට ලැබෙන ප්‍රතිදානය / **output** වන්නේ,

- i) **[1,0,1,0,0,0,0,0]**            ii) **[1,0,1,0,0,0,0,0]**            iii) **[1,0,1,0,0,5,0,0]**  
iv) **[1,0,1,0,1,0,1,1]**            v) **[0,0,0,0,1,0,1,0]**

**311. 100** අගය ආදානය / **input** කළ විට ලැබෙන ප්‍රතිදානය / **output** වන්නේ,

- i) **[0,1,1,0,0,1,0,0]**            ii) **[0,1,1,0,0,1,0,0]**            iii) **[1,1,0,0,1,0,0]**  
iv) **[1,1,1,0,0,1,0,0]**            v) **[1,1,0,0,1,0,0]**

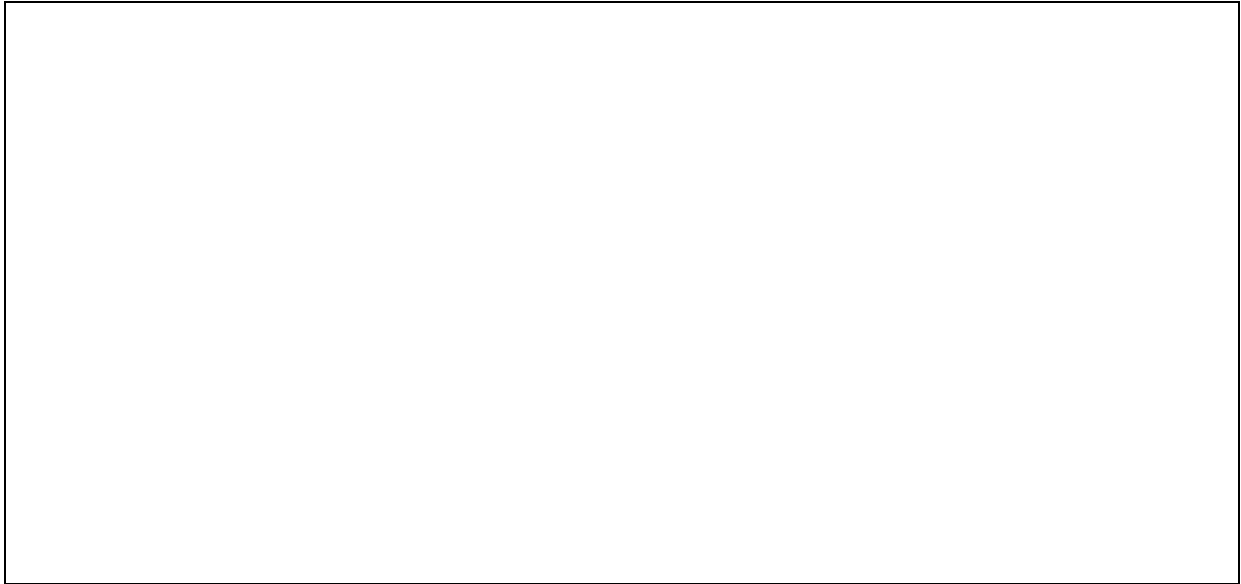
**312. [1,0,1,1,1,0,0,1]** ප්‍රතිදාන / **output** ලබා ගැනීමට ආදානය / **input** කළ යුතු අගය වන්නේ,

- i) **190**            ii) **186**            iii) **185**            iv) **157**            v) **None**

**313. 256** ආදානය කළ විට සිදුවන්නේ,

- i) **[1,0,1,0,0,1,0,1]**            ii) **[1,1,1,1,1,1,1,1]**            iii) **[1,1,1,1,1,1,1]**  
iv) **[0,1,1,1,0,1,0,0]**            v) **Error**

**314.** ඉහත කේතය කේතනයට පෙර සකසන ලද ගැලීම් සටහන පහත ඇද දක්වන්න.



**315.** ත්‍රිකෝණයක වර්ගපලය / area of triangle සෙවීම සඳහා සිසුන් පස් දෙනෙකු විසින් සකස් කරන ලද පයිතන් කේත පහත දක්වා ඇත. **Input** ලෙස පාදයේ දිග(a) ලම්බ උස ලබා දේ(b).

ඒ අතරින් නිවැරදි කේතය / කේත වන්නේ,

- i) `a = int(input ( ) )`  
`b = int(input ( ) )`  
`Print (a * b ** 2)`
- ii) `a = int(input ( ) )`  
`b = int (input ( ) )`  
`print (a * b * 1/2)`
- iii) `a = int(input())`  
`b = int(input ( ) )`  
`print(a * b * 0.5)`
- iv) `a = intput()`  
`b = intput()`  
`print(a * b *0.5)`

v) ii , iii දෙකම නිවැරදි

>>> පහත පයිතන් කේතය (python Code) කේතය සලකන්න. එමගින් 316 – 319 දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
no = [ 1 , 4 , 9 , 8 , 10 ]
c , m = 1 , no[ 0 ]
while _____ : # space A
    if no[ c ] > m:
        m = no[ c ]
    c = c + 1
print( m )
```

**316.** No නම් වූ list එකේ විශාලතම අගය print කර ගැනීමට A හිස්තැනට යෙදිය යුත්තේ,  
 i) `m > no[m]`      ii) `m > no[c]`      iii) `no[c] > m`      iv) `no[c] >= m`      v) `len(no) > c`

**317.** No නම් වූ list එකේ කුඩාම අගය print කර ගැනීමට if වෙනස් කළ යුත්තේ,  
 i) `m > no[m]`      ii) `m < no[c]`      iii) `no[c] < m`      iv) `no[c] >= m`      v) No one

**318.** A හිස්තැනට 309 හි iv පිළිතුර යොදන ලදී. එවිට ලැබෙන ප්‍රතිදානය / output වන්නේ,  
 i) 1      ii) 4      iii) 9      iv) 8      v) 10

319. ඉහත කේතයට අදාළව ගැලීම් සටහනක් අදින්න. ව්‍යාජ කේතයද ලියන්න.

320. පහත code එකෙහි output එක වන්නේ,

```
a = 'Dushan'
b = 19
```

```
print(' { } is { } years Old '.format( b , a ) )
```

- i) Dushan is 19 years old      ii) Dushan is 19 years old      iii) 19 is Dushan years old format  
iv) 19 is Dushan year Old      v) 19 is Dushen years Old

321. පහත code එකෙහි output එක වන්නේ,

```
a = 'Dushan'
b = 19
```

```
print(' { } is { } years /n old '.format( b , a ) )
```

- i) Dushan is 19 years old      ii) Dushan is 19 years old      iii) 19 is Dushan years /n old  
iv) 19 is Dushan year Old      v) 19 is Dushan years Old

322. පහත code එකෙහි output එක වන්නේ,

```
a = 'Dushan'
b = 19
```

```
print(' { } is { } years \n old '.format( b , a ) )
```

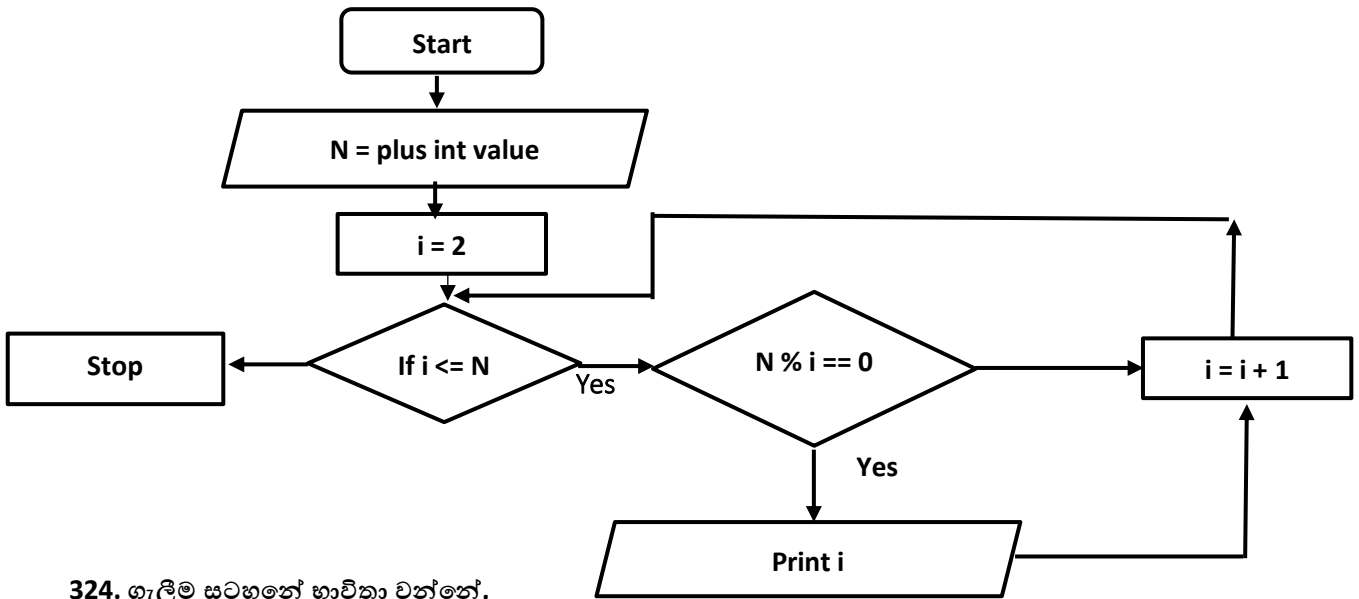
- i) Dushan is 19 years old      ii) Dushan is 19 years old      iii) 19 is Dushan years /n old  
iv) 19 is Dushan year old      v) old 19 is Dushan year

323. පහත code එකෙහි output එක වන්නේ,

```
import math
c = math.sqrt (100)
print(c)
```

- i) 10.0      ii) 10      iii) 10000      iv) 10000.0      v) Error

>>> පහත Flow Chart එක සලකන්න. එමගින් 324 – 325 දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.



324. ගැලීම් සටහනේ භාවිතා වන්නේ,

- i) අනුක්‍රමණය පමණි. / Only Sequence
- ii) අනුක්‍රමණය හා තේරීම පමණි. / Only Sequence and Selection
- iii) අනුක්‍රමණය හා පුනර්කරණය පමණි. / Only Sequence and iteration
- iv) තේරීම හා පුනර්කරණය පමණි. / Only Selection and Iteration
- v) අනුක්‍රමණය , තේරීම හා පුනර්කරණය පමණි. / All

325. '8' ආදානය / input කළ විටදී සිදුවන දෑ නිරූපණය කරන පහත නිවැරදි ප්‍රකාශය වන්නේ,

- i) i හි අගය 8 ට සමාන වන විට එය නවතී.
- ii) එය ප්‍රතිදානයක් ලෙස 8 මුද්‍රණය කරවයි
- iii) එය ප්‍රතිදානයක් ලෙස 6 මුද්‍රණය කරවයි
- iv) එය 1 – 8 තෙක් සියලු ඉරට්ටේ integer අගයන් මුද්‍රණය කරවයි
- v) එය 1 – 6 තෙක් සියලු ඉරට්ටේ integer අගයන් මුද්‍රණය කරවයි

>>> පහත පයිතන් කේතය (python Code) කේතය එමගින් 326 – 328 දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```

Letters = ['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z']
K = input (" Enter a Letter : ")
print ( Letters.index( K ) + 1)
    
```

326. ඉහත කේතය සඳහා 'h' අකුර ආදානය කරන ලදී. එවිට ප්‍රතිදානය / output වන්නේ,

- i) 8.0                      ii) 7                      iii) 9                      iv) 8                      v) Error

327. ඉහත කේතය සඳහා 'Q' අකුර ආදානය කරන ලදී. එවිට ප්‍රතිදානය / output වන්නේ,

- i) 17                      ii) 17.0                      iii) 18                      iv) No output                      v) Error

328. ඉහත කේතයේ දී සමහර English අක්ෂර ආදානයන් වලදී [input] Error පණිවිඩ ලැබෙන බව අමල් පවසයි. එය පරීක්ෂා කළ ගුරුවරයාද එයට එකඟ වූ අතර එය නිවැරදි කිරීමට නම්,

- i) print (Letter.index( K ) + 1) එක , print (Letter.index( K ) + '1') ලෙස වෙනස් කිරීම.
- ii) Letters නම් වූ List එක Tuple එකක් ලෙස සකස් කිරීම.
- iii) K = input ("Enter a Letter : ") එක, K = input ("Enter a Letter : ").lower() ලෙස වෙනස් කිරීම.
- iv) K = input ("Enter a Letter : ") එක, K = Str(input ("Enter a Letter : ")) ලෙස වෙනස් කිරීම.
- v) ඉහත කිසිවක් නොවේ.

**329.** අමල් යනු පාසලෙහි ICT විෂය හදාරන දක්ෂ සිසුවෙකි. ඔහු Python Programming Language එක භාවිතා කර මෘදුකාංගයක් නිපදවීමට උත්සහ දරයි. එවිට ඔහුට ගැටලුවක් පැන නැගිණි. එනම් පරිශීලකයා ආදානය කරන / දත්ත / ගබඩා කිරීම සම්බන්ධයෙනි. පරිශීලකයා ආදානය කරන දත්ත නියත වශයෙන්ම පරිගණක ද්විතීයික මතකයේ ගබඩා කිරීමට අවශ්‍ය විය. මේ ගැටලුවට ලබා දිය හැකි සරලම විසඳුම වන්නේ,

- i) List භාවිතා කර දත්ත ගබඩා කිරීම.
- ii) .txt file භාවිතා කර දත්ත ගබඩා කිරීම.
- iii) SQL භාවිතා කොට දත්ත ගබඩාවක් නිපදවා එය තුළ දත්ත ගබඩා කිරීම.
- iv) Tuple භාවිතා කොට වෙනස් කළ නොහැකි පරිදි දත්ත ගබඩා කිරීම.
- v) අන්තර්ජාලයේ Sever Computer තුළ දත්ත ගබඩා කිරීම.

**330.** Python File භාවිතයේදී, කියවීමට පමණක් හැකි වන පරිදි ideapack. Txt නම් වූ ගොනුවක් විවෘත කිරීමට අදාළ පයිතන් වගන්තිය වන්නේ.

- i) `open = f ( ' ideapack.txt ' , ' r ' )`
- ii) `open = open ( ' ideapack.txt ' , ' r ' )`
- iii) `f = open ( ' ideapack.txt ' , ' w+ ' )`
- iv) `e = open ( ' ideapack.txt ' , ' r ' )`
- v) `f = open ( ' ideapack.txt ' , ' a ' )`

**331.** `File = open ( ' ideapack.txt ' , ' a ' )` ලෙස කේතනය මගින් ගොනු විවෘත කිරීමේ නිවැරදි අරමුණ වඩාත්ම හොඳින් පැහැදිලි වන්නේ,

- i) ගොනුව තුළ ඇති දත්ත කියවීමට හා ලිවීමට යන දෙකම සිදුකළ හැකිවන පරිදි ගොනු විවෘත කිරීම.
- ii) ප්‍රමුඛතාවය ලිවීමට වූවත් යම් වෙලාවක් වලදී කියවීමට ද හැකිවන ලෙස ගොනු විවෘත කිරීම.
- iii) දැනට පවතින දත්ත මැකීමකින් තොරව නව දත්ත ලිවීමට හැකිය.
- iv) ප්‍රමුඛතාවය කියවීම වූවත් යම් වෙලාවක් වලදී ලිවීමට ද හැකිවන ලෙස ගොනු විවෘත කිරීම.
- v) ඉහත කිසිවක් නොවේ.

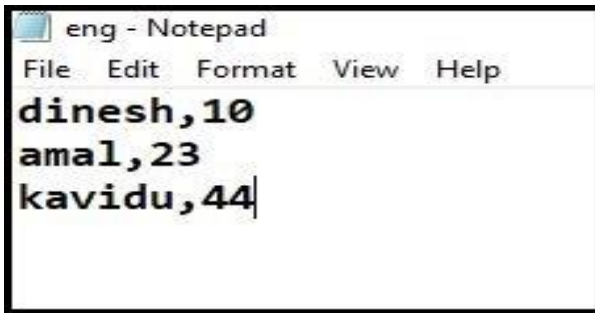
**332.** `File = open ( ' ideapack.txt ' , ' w ' )` හා `File = open ( ' ideapack.txt ' , ' w+ ' )` අතර වෙනස වන්නේ,

- i) W කියවීමට පමණි. W+ලිවීමට ද කියවීමටද හැකිය. දෙකම අලුත් ගොනුවක් නිර්මාණය කරගනී.
- ii) W ලිවීමට පමණි. W+ලිවීමට හා කියවීම කළ හැකිය. අලුත් ගොනුවක් නිර්මාණය කරගනී.
- iii) W මගින්, ලිවීම සිදු කළ නොහැකි අක්ෂර W+ මගින් ලිවීම කළ හැකිය.
- iv) W මගින්, නිපදවා ඇති ගොනුවක ලිවීම කරයි. මගින් අලුත් ගොනුවක් නිපදවා ලිවීම කරයි.
- v) ඉහත කිසිවක් නොවේ.

**333.** `File = open ( ' ideapack.txt ' , ' r ' )` හා `File = open ( ' ideapack.txt ' , ' r+ ' )` අතර වෙනස වන්නේ,

- i) r කියවීම පමණක් වන අතර r+ මගින් කියවීම ලිවීම දෙකම කළ හැකිය. r+ මගින් නවගොනුවක් නිර්මාණය කරයි.
- ii) r කියවීම පමණක් වන අතර r+ මගින් කියවීම ලිවීම දෙකම කළ හැකිය. r+මගින් නවගොනුවක් නිර්මාණය නොකරයි.
- iii) r මගින් කියවීම සිදු කළ නොහැකි අක්ෂර r+ මගින් කියවා ගත හැකිය.
- iv) කිසිදු වෙනසක් නොපවතී.
- v) ඉහත කිසිවක් නොවේ.

>>> පහත දක්වා ඇත්තේ නම් වූ ගොනුවක දත්ත ගබඩා වී ඇති ආකාරය වේ. ඒ ඇසුරෙන් **334** සිට **337** දක්වා ප්‍රශ්න වලට පිළිතුරු සපයන්න. [ ගොනුවේ නාමය **eng . txt** ]



**334.** පහත පයිතන් කේතය ධාවනයෙන් ලැබෙන ප්‍රතිදානය / **output** වන්නේ,

```
File = open ('eng .txt' , 'r')
A = File.readlines() ;
print (A) ; File.close()
```

- i) ['dinesh,10\n', 'amal,23\n', 'kavidu,44']
- ii) ['dinesh,10', 'amal,23', 'kavidu,44']
- iii) [[dinesh 10], [amal,23], [kavidu,44]]
- iv) dinesh,10
- v) dinesh,10  
amal,23  
kavidu,44

**335.** පහත පයිතන් කේතය ධාවනයෙන් ලැබෙන ප්‍රතිදානය / **output** වන්නේ,

```
File = open ('eng .txt' , 'r')
A = File.readline()
print (A)
B = File.readline()
print (B)
File.close()
```

- i) ['dinesh,10\n', 'amal,23\n', 'kavidu,44']
- ii) ['dinesh,10', 'amal,23', 'kavidu,44']
- iii) ['dinesh 10' , 'amal,23']
- iv) dinesh,10 | amal,23
- v) dinesh,10  
  
amal,23

**336.** පහත පයිතන් කේතය ධාවනයෙන් ලැබෙන ප්‍රතිදානය / **output** වන්නේ,

```
File = open ('eng .txt' , 'r')
A = File.read()
print (A)
File.close()
```

- i) ['dinesh,10\n', 'amal,23\n', 'kavidu,44']
- ii) ['dinesh,10', 'amal,23', 'kavidu,44']
- iii) [[dinesh 10], [amal,23], [kavidu,44]]
- iv) dinesh,10
- v) dinesh,10  
amal,23  
kavidu,44

**337.** ඉහත **eng.txt** ගොනුවටම **kalana,78** ලෙස දත්ත **3** වම පහළින් ලිවීමට / **write** කේතය වන්නේ,

i) **File = open (' eng.txt ', ' r ')**  
**File.write ('kalana,78')**  
**File.close()**

ii) **File = open (' eng.txt ', ' w ')**  
**File.write('kalana,78\n')**

iii) **File = open (' eng.txt ', ' a ')**  
**File.write ('\nkalana,78')**  
**File.close()**

iv) **File = open (' eng.txt ', ' r+ ')**  
**File.write ('kalana,78\n')**

v) ii) and iii)

>>> පහත ව්‍යාජ / **Pesodu Code** කේතය සලකා **338** හා **339** ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

**Begin of Programe**

**Sum = 0**

**For number = -10 To 11 Step 2**

**If number is greater than 0 Then**

**sum = sum + number**

**End if**

**Next count**

**Display sum**

**End**

**338.** ඉහත ව්‍යාජ කේතයෙහි ප්‍රදානය / **output** වන්නේ,

i) **0**

ii) **10**

iii) **6**

iv) **30**

v) **50**

**339.** ඉහත ව්‍යාජ කේතය සඳහා සුදුසු ව්‍යාජ / **Pesodu Code** කේතය සලකා ගැලීම් සටහන / **Flow chart** අඳින්න.



>>> පහත ව්‍යාජ / Pesodu Code කේතය සලකා 340 සිට 342 දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```

Start Progrme
M = Input Subject Marks
If M < 0 or M > 100 Then
    Print "Marks not Valid"
Else
    If M > 75 Then
        Print "You are very clever"
    Else if M > 40 Then
        Print "Pay attention toward your studies"
    Else
        Print "Work much hard"
End Programe
    
```

A Line  
B Line  
C Line  
D Line

340. ඉහත ව්‍යාජ කේතයේ දැක්වෙන ගැලීම පාලන ව්‍යුහය වන්නේ,  
 i) Sequence only                      ii) Sequence and Selection only                      iii) Selection only  
 iv) Sequence and Iteration only                      v) Sequence, Selction and Iteration

341. බිංදුවට වඩා අඩු අගයක් ඇතුළත් කළ විට එය ප්‍රතික්ෂේප කරන කාර්යය සිදුකරන ඡේද්‍රිය/ Line වන්නේ,  
 i) A                      ii) B                      iii) C                      iv) D                      v) A plus B

342. ඉහත ඇල්ගොරිතමය සම්බන්ධයෙන් ඇති පහත ප්‍රකාශ අතරින් නිවැරදි ප්‍රකාශ / ප්‍රකාශය වන්නේ,  
 i) 0 ට අඩු අගයක් ආදානය / input කළ විට වැඩසටහන නතර වී නැවත ධාවනය සමහර විට විය හැක.  
 ii) 75 අගය ආදානය / input කළ විට Pay attention toward your studies ප්‍රතිදානය / Print වේ.  
 iii) 40 අගය ආදානය / input කළ විට Pay attention toward your studies ප්‍රතිදානය / Print වේ.  
 iv) 40 අගය ආදානය / input කළ විට Work too much ප්‍රතිදානය / Print වේ.  
 v) i , ii , iii නිවැරදි

>>> පහත Python කේතය සලකා 343 සිට 346 දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```

A = int (input ("Enter Input 1 :"))
B = int (input ("Enter Input 2 :"))
C = A * B
if C >= 1 :
    C = 1
else:
    C = 0
print (C)
    
```

343. ඉහත කේතයේ ආදාන හා ප්‍රතිදාන input / output වලට සමාන ලෙස ක්‍රියා කරන ද්වාර පරිපතය / Logic Gate වන්නේ, ආදාන සඳහා 0 හා 1 පමණක් ආදානය කරන බව සලකන්න.  
 i) XOR                      ii) NAND                      iii) NOR                      iv) AND                      v) NOT

344. ඉහත කේතයම OR ආකාරයට හැසිරෙන ලෙස සකස් කරන්නේ නම්, ඉහත කේතයේ සිදු කළ යුතු වෙනස් කම වන්නේ,  
 i) if C >= 1 → if C < 1                      ii) C = 1 → C = 0                      iii) C = A \* B → C = A + B  
 iv) C = A \* B → C = A - B                      v) ඉහත කිසිවක් නොවේ.

**345.** ඉහත කේතයම XOR ආකාරයට හැසිරෙන ලෙස සකස් කරන්නේ නම්, ඉහත කේතයේ සිදු කළ යුතු වෙනස්කම වන්නේ,

- `if C >= 1 → if C == 1 & C = A * B → C = A + B`
- `if C >= 1 → if C <= 1 & C = A * B → C = A - B`
- `C = 1 → C == 0 & C = 0 → C == 1`
- `if C >= 1 → if C == 1 & C = A * B → C = A / B`
- `if C >= 1 → if C <= 1`

**346.** ඉහත පයිතන් කේතයට නියමිත ව්‍යාජ කේතය ලියන්න.

>>> පහත ව්‍යාජ / Pesodu Code කේතය සලකා 347 හා 348 ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

[ පහත දක්වා ඇත්තේ සිසුවෙකු විසින් සකස් කරන ලද Mobile Phone Number වැඩ සටහනකි. ]

```
C = { "KAMAL" : '079457860' , "DILAN" : '0795644330' , "DUSHAN" : '0796655665' }
A = input("Search Name : ")
A = A.upper() #3rd Line
print("Phone Number : " + C[A] )
```

**347.** ඉහත කේතයේ තුන් වන ඡේදියේ / 3<sup>rd</sup> Line අවශ්‍යතාව වන්නේ,

- Int ලෙස පැමිණෙන අගය String බවට පත් කිරීම,
- පයිතන් Simple Letter , Capital Letters හේදයක් පවතින නිසා.
- අත්‍යවශ්‍යම කේත ඡේදියක් නොවුණත් වඩා ආරක්ෂාව සඳහා යොදා ඇත.
- A සඳහා Int වර්ගයේ අගයන් ඇතුළු කරගොත් ඒ සඳහා පිළියමක් ලෙස.
- ඉහත සියලුම හේතුන් නිසා එය යොදා ගනියි.

**348.** Dushan ලෙස ආදානය / input කළ විට ප්‍රතිදානය / output වන්නේ,

- 0796655665
- '0796605665'
- 079457860
- '079457860'
- Error

>>> පහත Python කේතය සලකා 349 සිට 353 දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
A = input(Enter Your Full Name : )
B = A.split(' ') # There is a Space between Bracks (*Space')
for k in range(len(B) - 1) :
    C = B[k] . upper()
    print (C[0], end = ' . ')
print(B[-1].upper())
```

**349.** කේතයේ Syntax Error පවතී. එය නිවැරදි කිරීමට නම් එය වෙනස් කළ යුත්තේ පහත කුමන ආකාරයෙන් ද

- `A = Str( input(EnterYour Full Name : ) )`
- `A = int(input('Enter Your Full Name : '))`
- `A = input.display('Enter Your Full Name : ')`
- `A = Str(input('Enter Your Full Name : '))`
- `A = input('Enter Your Full Name : ')`

**350.** වැරදි නිවැරදි කරන ලද ඉහත කේතයට name ආදානය [input] කළ විට ලැබෙන ප්‍රතිදානය [output] වන්නේ,

- නමෙහි කොටස් වල මූලකරු හා කොටස් ගණන
- මූලකරු සමග නම
- අකුරු සියල්ලම **Capital** අකුරු බවට පත් කරන ලද නම
- මූලකරු සමග , සම්පූර්ණ **Capital** අකුරු සහිත නම
- ඉහත කිසිවක් නොවේ

**351.** මෙම නාමය ආදානය [input] කළ පසු ලැබෙන ප්‍රතිදානය [output] වන්නේ,  
**Ranbanda Arachchige don Kalana Sankalpa**

- R.A.D.K.S
- R.A.d.K.Sankalpa
- R.A.D.K.SANKALPA
- R.A.D.Kalana Sankalpa
- 'R.A.D.KALANA SANKALPA'

**352.** මෙය ආදානය / input කළ පසු ලැබෙන ප්‍රතිදානය / output වන්නේ,  
**1902 3456 277B 7777**

- 1.3.2.7777
- 1.3.2.7
- 1.3.2.7.777
- 1.3.2.77.77
- Error

**353.** මෙවැනි කේතයන්ගේ භාවිතයක් වන්නේ,

- OS සඳහා
- Form SW සඳහා
- Virus Guard සඳහා
- System SW සඳහා
- None

**354.** ශ්‍රිත / Functions භාවිතයේ වාසියක් වන්නේ,

- පහසුවෙන් කේතනයේ යෙදිය හැකිය.
- කේතනය යන කාලය අවම කර ගත හැකිය.
- කේතය සඳහා යන පේළි ප්‍රමාණය අඩු කර ගත හැකිය.
- එකම කේත කොටස නැවත නැවත ලිවීමට සිදුවන අවස්ථා නැති කර ගත හැකිය,
- ඉහත සියල්ලම නිවැරදි.

**355.** Functions එකක් සාදන නිවැරදි ආකාරය වන්නේ,

- def #A:
- def g :
- def @G :
- def R ;
- None

>>> පහත පයිතන් කේතය සලකා 356 හා 357 ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
A = input('Enter :')
```

```
B = A.split('.')
```

```
C = int(A[0])
```

```
if C >= 240 : print(' E')
```

```
elif C >= 224 : print(' D')
```

```
elif C >= 192 : print(' C')
```

```
elif C >= 128 : print(' B')
```

```
elif C >= 0 : print(' A')
```

**356.** ඉහත කේතයේ අරමුණ / කාර්යය වන්නේ,

- ඇතුළත කරන විඛාග ලකුණු [Exam Marks] වලට අදාළ පන්තිය ලබාදීම
- ඇතුළත් කරන අගයේ විශාලත්වය අනුව ඉංග්‍රීසි අකුරු ලබා දීම.
- ඇතුළත් කරන ලද වචනයේ අක්ෂර ගණනට අදාළව පන්ති ලබා දීම.
- Port Numbers ඇතුළු කළ විට එයට අදාළ පන්තිය / Class ලබා දීම.
- IP addresses ඇතුළු කළ විට එයට අදාළ පන්තිය / Class ලබා දීම.

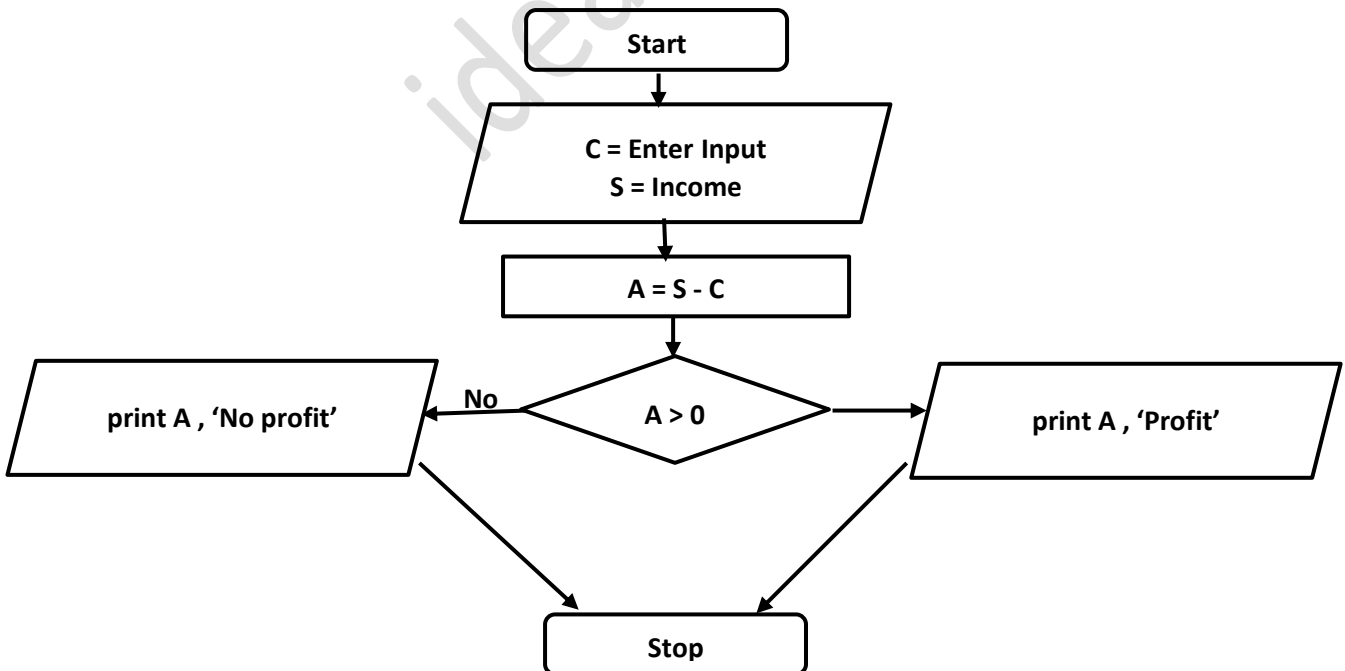
357. ඉහත කේතයට ව්‍යාජ කේතය ලියන්න.

358. පහත කේතය සම්බන්ධයෙන් ඇති ප්‍රකාශ වලින් නිවැරදි ප්‍රකාශය වන්නේ,

```
"""# start of the code
X = int(input('Enter val : '))
print('X') """
```

- i) ආදානය කරන / input value අගය ප්‍රතිදානය / output වේ.
- ii) X ලෙස ප්‍රතිදානය / output කරයි.
- iii) ප්‍රතිදානයන් / outputs ලබා නොදෙන අතර ආදාන / inputs ලබා ගැනීම අවකාශය සලසයි.
- iv) ආදානය / input හෝ ප්‍රතිදාන / output ලබා දීම හෝ ලබා ගැනීම කළ නොහැකිය.
- v) දෝෂ පණිවිඩ ලබා දේ / Error Messagers.

>>> පහත Flow Chart / ගැලීම සටහන සලකා 359 සිට 360 දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.



359. ඉහත ගැලීමට ආදාන ලෙස පිළිවලින් 400, 1000 ආදානය කරන ලදී. එවිට ප්‍රතිදානය වන්නේ,

- i) 600 profit
- ii) 600 No profit
- iii) -600 profit
- iv) -600 No profit
- v) Profit

360. ඉහත කේතයට අදාළ පයිතන් කේතය වන්නේ,

```
S = int(input (' Enter Your Cost : '))
C = int(input (' Enter Your Income : '))
A = S - C
if A > 0 :
    print('No Profit' + Str(A))
else :
    print('Profit' + Str(A))
```

```
S = input (' Enter Your Cost : ')
C = input (' Enter Your Income : ')
A = S - C
if A > 0 :
    print('No Profit' + Str(A))
else :
    print('Profit' + Str(A))
```

```
S = int(input (' Enter Your Cost : '))
C = int(input (' Enter Your Income : ')),
A = S - C
if A > 0 :
    print('No Profit' + Str(A))
else :
    print('Profit' + Str(A))
```

```
S = int(input (' Enter Your Cost : '))
C = int(input (' Enter Your Income : '))
A = S - C
if A > 0 :
    print ('No Profit' + A)
else :
    print ('Profit' + A)
```

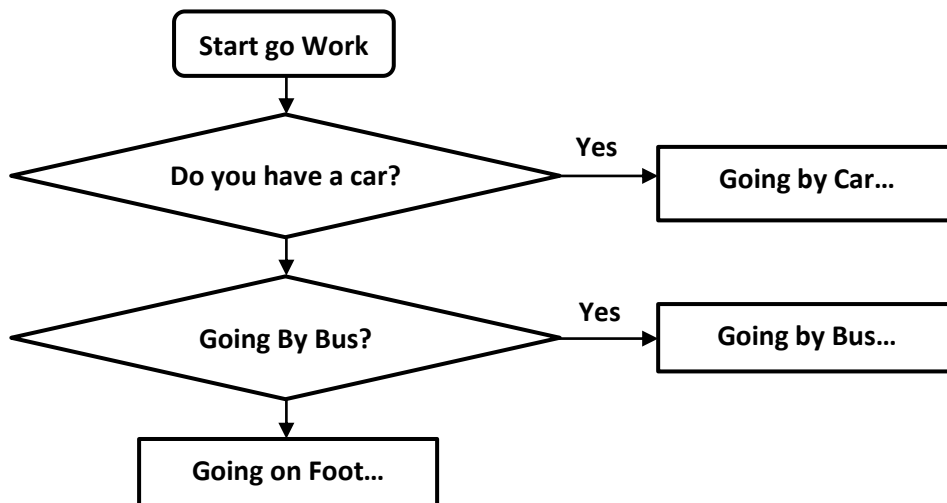
v) ALL Wrong

361. පහත කේතයට ලබා දෙන යම් යම් ආදාන සඳහා දෝෂ පණිවිඩ ලබා දේ. එයට හේතු විය හැක්කේ,

```
A = int (input ('Enter a Value: '))
B = int (input ('Enter a Value: '))
C = A / B
print(C)
```

- i) Variables Capital Letters වලින් ආරම්භ කර තිබීම නිසා.
- ii) බිංදුව [ Zero], A සඳහා ආදානය / input කළ විට පැමිණේ.
- iii) බිංදුව [ Zero], B සඳහා ආදානය / input කළ විට පැමිණේ.
- iv) සමහර ආදාන වලට පමණක් නොව සියලුම ආදාන සඳහා පණිවිඩ ලබා දේ.
- v) කිසිම ආදානයක් සඳහා පණිවිඩ ලබා නොදේ.

362. පහත දැක්වෙන ඇල්ගොරිතමය වඩාත් හොඳින් පැහැදිලි වන්නේ,



- i) කාරයක් නැතිනම් වැඩපලට බසයෙන් ගමන් කරනා ලෙසයි.
- ii) කාරයක් හා පයිත් ගමන් කිරීමටද නොහැකි නම් පොදු බස් සේවාවක් භාවිතා කරන ලෙස.
- iii) කාරයක් හෝ පොදු බස් සේවා නැති නම් පයිත් ගමන් කරනා ලෙස
- iv) කාරයක් හෝ පයිත් ගමන් කිරීමටද නොහැකි නම් පොදු බස් සේවාවක් භාවිතා කරන ලෙස.
- v) කාරයක් නැතිනම් පාසලට පොදු බස් සේවාව මගින් ගමන් කරනා ලෙස

363. පරිගණක භාෂා විකාෂණය සම්බන්ධයෙන් නිවැරදි ප්‍රකාශය වන්නේ,

- i) **Asembly Languages / Machine Languages / High Level Languages**
- ii) **Machine Languages / Asembly Languages / High Level Languages**
- iii) **High Level Languages / Asembly Languages / Machine Languages**
- iv) **High Level Languages / Asembly Languages / C# Languages**
- v) **C# Languages / C++ Languages / Java Languages**

364. කට වහරේ හි භාෂාවක් හා පරිගණක ක්‍රමලේඛණ භාෂාවක් අතර වෙනස වන්නේ,

- i) කට වහර භාෂාවල වචන භාවිතා කරන අතර පරිගණක ක්‍රම ලේඛන භාෂාවල ඉලක්කම් භාවිත වේ.
- ii) කට වහර භාෂාවල හා පරිගණක භාෂා අතර කිසිදු වෙනස්කමක් නැත.
- iii) පරිගණක භාෂා පරිගණකයේ මතකයේ පමණක් පවතින අතර කට වහර භාෂා මිනිසාගේ මතකයේ පමණක් ඇත.
- iv) මිනිසා සමග සන්නිවේදනයට කට වහර භාෂාද පරිගණකයට විදාන ලබා දීමට පරිගණක භාෂා භාවිතා කරයි.
- v) ඉහත iii හා iv දෙකම නිවැරදි.

365. පහත කේතයේ / Code ප්‍රතිදානය / output වන්නේ,

```
X = 10
while X > 2:
    X = X - (2+1)
    print(X, end = '')
```

- i) 7 6                      ii) 1 4 7                      iii) 2 7 1                      iv) 7 4 1                      v) Error

366. පහත කේතයේ ප්‍රතිදානය වන්නේ,

```
def Func (Z, Y = []):
    for i in range( Z ):
        Y . append( i )
    return Y
print( Func (3, [ 1 , 2 , 3 ] ) )
```

- i) [1,2,3,3]                      ii) [1,2,3,0,1,2]                      iii) [1,2,3,4,5,6]                      iv) [1,2,3,0,1,3]                      v) [0,1,2,1,2,3]

367. Function එකක් සම්බන්ධයෙන් පහත දක්වා ඇති කරුණු වලින් නිවැරදි ප්‍රකාශ / ප්‍රකාශය වන්නේ,

- a. **Function** යනු පයිතන් භාෂාවේ පමණක් ඇති අංගයකි.
- b. ක්‍රමලේඛකයාට අවශ්‍ය පරිදි **Function** සාදා ගත හැකිය
- c. **Function** එකක් තුළදී තවත් **Function** එකකට **Call [Function Calling]** කළ හැකිය.

- i) a පමණි.                      ii) a, b පමණි.                      iii) a , b , c සියල්ලම.                      iv) b , c පමණි.                      v) සියල්ල.

>>> ත්‍රිකෝණ සංඛ්‍යා / **Triangular Number** යනු ත්‍රිකෝණයක් නිර්මාණයට දායක වන සංඛ්‍යා වේ. **1 – 10** දක්වා ත්‍රිකෝණ සංඛ්‍යා ප්‍රතිදානයට වැඩසටහනක් නිර්මාණය අවශ්‍ය වී ඇත.

**368.** 1 – 10 අතර පවතින ත්‍රිකෝණ සංඛ්‍යා වන්නේ,

- i) 1,3,6,9                      ii) 0,1,2,3,9,10                      iii) 1,4,9                      iv) 1,3,6,10                      v) 2,3,4,5

**369.** 1 – 100 අතර ත්‍රිකෝණ සංඛ්‍යා ප්‍රතිදානය කිරීමට අදාළ නිවැරදි පයිතන් කේතය වන්නේ,

```
a , b = 0 , 0
while b < 100 :
    print(b)
    a = a + 1
    b = b + a
```

```
a , b = 1 , 0
while b < 100 :
    print(b)
    a = a + 1
    b = b + a
```

```
a , b = 1 , 1
while b < 100 :
    print(b)
    a = a + 1
    b = b + a
```

```
a , b = 1 , 1
while a < 100 :
    print(b)
    a = a + 1
    b = b + a
```

v) සියල්ලම නිවැරදි.

**370.** ඉහත කේතය සඳහා සුදුසු ගැලීම් සටහනක් අදින්න.

**371.** පහත කේතය ප්‍රකාශ අතරින් සත්‍ය ප්‍රකාශය වන්නේ,

```
a = 0
while a < 10:
    a = a + 1
    print(a**2)
```

- i) 0 – 10 දක්වා වර්ග සංඛ්‍යා ප්‍රතිදානය කරයි.  
 ii) 0 – 10 අතර ද්විමය සංඛ්‍යා ප්‍රතිදානය කරයි.  
 iii) 0 – 100 අතර වර්ග සංඛ්‍යා ප්‍රතිදානය කරයි.  
 iv) 1 – 10 දක්වා ගණන සංඛ්‍යා ප්‍රතිදානය කරයි.  
 v) 0 – 100 අතර ත්‍රිකෝණ සංඛ්‍යා ප්‍රතිදානය කරයි.

372. පහත කේතයේ ප්‍රතිදානය / output වන්නේ,

```
a = 0
while a < 10:
    if a % 2 == 0 or a % 3 == 1 :
        print(a, end = " ")
    a = a + 1
```

- i) 0124678      ii) 1,2,4,6,7,8      iii) 013678      iv) 12345      v) 346789

373. පහත කේතයේ ප්‍රතිදානය / output වන්නේ,

```
for i in range( 5 ) :
    i = 5
    print(i, end = ',')
```

- i) 0,1,2,3,4      ii) 01234      iii) 5,5,5,5,5 ,      iv) 55555      v) Nothing

374. පහත කේතයේ ප්‍රතිදානය / output වන්නේ,

```
A = 0
while not(A > 5) :
    print(A , end = " ")
    A = A + 1
```

- i) 012345      ii) 01234      iii) 0123456      iv) 02468      v) 13579

375. පහත කේතයේ ප්‍රතිදානය / output වන්නේ,

```
F = [ ' b ' , ' i ' , ' m ' , ' s ' , ' a ' , ' r ' , ' a ' ]
E = " . join ( F )
print ( E )
```

- i) b,i,m,s,a,r,a      ii) bimsara      iii) bimsar      iv) [bimsara]      v) Error

376. පහත කේතයේ ප්‍රතිදානය / output වන්නේ,

```
F = [ ' b ' , ' i ' , ' m ' , ' s ' , ' a ' , ' r ' , ' a ' ]
E = ' - ' . join ( F )
print ( E )
```

- i) b,i,m,s,a,r,a      ii) b-i-m-s-a-r-a      iii) b-i-m-s-a-r-a-      iv) [bimsara]      v) Error

377. ආදානය [input] ලෙස යම් කිසි වර්ෂයක් ගෙන එය දීර්ඝ වසරක් [Leap Year] නම් 'Leap' ලෙසද නැතිනම් 'No Leap' ලෙසද ප්‍රතිදානය [output] කිරීමට අදාළ කේත 5 ක් පහත දැක් වේ. ඒ අතරින් නිවැරදි කේතය වන්නේ,

i) `Y = int(input ('Enter Year : '))`  
`A = [ ' Leap ' , ' No Leap ' ]`  
`print( A [ Y % 4 ] )`

ii) `Y = int(input ('Enter Year : '))`  
`A = [ 'No Leap ' , ' Leap ' ]`  
`print( A [ Y % 4 ] )`

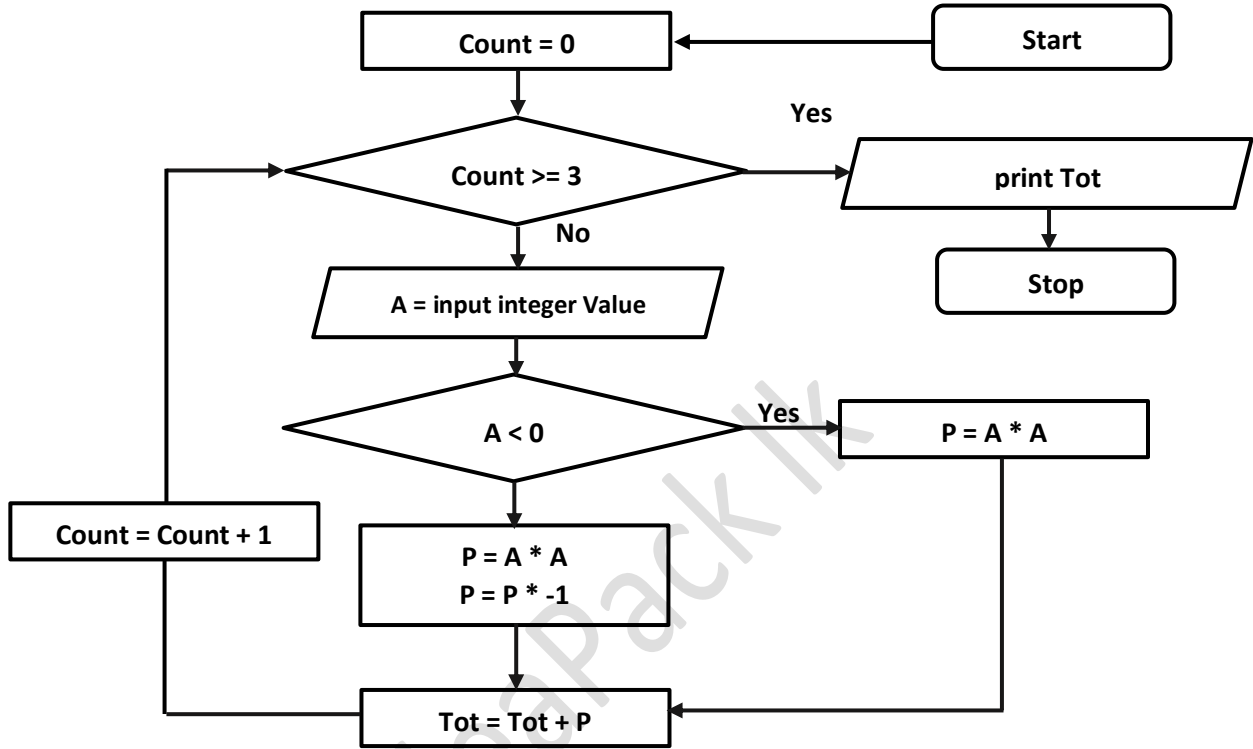


```
iii) Y = int(input ('Enter Year : '))
A = [ 'Leap ', ' Leap ' ]
print( A [ Y % 4 ] )
```

```
iv) Y = int(input ('Enter Year : '))
A = [ 'Leap ', ' Leap ' ]
print( Y [ A % 4 ] )
```

v) No one Correct

>>> පහත Flow Chart / ගැලීම් සටහන සලකා 378 සිට 381 දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.



378. ඉහත ඇල්ගොරිතමයේ / Algorithm ආදාන / input කීයක් ලබා ගන්නේ ද,

- i) 1
- ii) 2
- iii) 3
- iv) 4
- v) 5

379. ඉහත ඇල්ගොරිතමයට 4, -2, -8, 0, 9 පිළිවලින් ආදාන ලබා දෙන ලදී. එවිට ප්‍රතිදානය වන්නේ,

- i) 52
- ii) 72
- iii) 88
- iv) -48
- v) -29

380. ඉහත ඇල්ගොරිතමයට 5, 7, 8, 10, 4 පිළිවලින් ආදාන ලබා දෙන ලදී. එවිට ප්‍රතිදානය වන්නේ,

- i) -254
- ii) 0
- iii) 238
- iv) 137
- v) -138

381. ඉහත ගැලීම් සටහන අනුව සිසුවෙකු විසින් සකස් කරන ලද පයිතන් කේතයක් පහත දැක් වේ.

```
count,Tot,P = 0,0,0
while count > 3:
    A = int(input("Enter a Value : "))
    if A < 0 :
        P = A**2
    else:
        P = A**2
        P = P*-1
    Tot = Tot + P
    count = count + 1
print(Tot)
```

ඉහත කේතයට අනුව ඇල්ගොරිතමයේ එකදු හෝ ආදානයක් ලබා ගැලීමට අවකාශය ලබා නොදේ.මෙය නිවැරදි කර ගැනීමට නම්,

- i) පළමු ජේලියේ ඇති **count = 0** එක **count = 1** ලෙස වෙනස් කිරීම
- ii) **while >= 3** ය **while <= 3** ලෙස වෙනස් කිරීම
- iii) **P = A\*\*2** ය **P = A\*A** ලෙස වෙනස් කිරීම
- iv) **count = count + 1** ය **count = count - 1** ලෙස වෙනස් කිරීම
- v) කේතයේ කිසිදු දෝශයක් නැත.

>>> පහත Python Code සලකා 382 සිට 385 දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
A = input('Enter a Number :')
count = 0
for i in range(len(A)):
    B = int(A[i])
    if B % 2 == 0:
        count = count + 1
    else :
        count = count - 1
print(count)
```

382. ඉහත කේතයේ ආදාන ලෙස / input - '6777454664' ඇතුළත් කරන ලදී. එවිට ප්‍රතිදානය / output වන්නේ,

- i) 5777466446
- ii) 5
- iii) 6446647775
- iv) 2
- v) 1

383. ඉහත කේතයේ ආදානය සඳහා ඉලක්කම් එකකින් පමණ සමන්විත අගයක් ඇතුළත් කළ විට ප්‍රතිදානය ලෙස 0 කිසිවිටකත් ලබා නොදෙන බව සිසුවෙකු පවසයි. එය සමග ඔබ,

- i) එකඟ නොවේ
- ii) එකඟ වේ
- iii) එය නියතව සිදු නොවේ
- iv) එය අහඹු සිදු වීමකි
- v) ලබා දෙන ආදාන මත එය තීරණය වේ.

384. පහත ප්‍රකාශ අතරින් නිවැරදි ප්‍රකාශ වන්නේ.

- a. A නම් Variable එකේ ආදානය [input] කරන ලද අගය String ලෙස ගබඩා වේ.
- b. ආදානය කරන ලද අගයේ ඇති තනි ඉලක්කම් ගෙන එය ඉරට්ටේ නම් count Variable එකට 1 එකතු කිරීමද නැතිනම් -1 අඩු කිරීමද මෙහිදී සිදුවේ.
- c. A = input('Enter a Number :') ය A = int(input('Enter a Number: ')) ලෙස වෙනස් කිරීම මගින් වඩාත් පහසුවෙන් මෙම කේතයම නිවැරදිව ක්‍රියාත්මක කළ හැකිය.

- i) a පමණි.
- ii) c පමණි.
- iii) a , b පමණි.
- iv) a , c පමණි.
- v) කිසිවක් නොවේ.

385. ඉහත කේතයට අදාල ගැලීම් සටහන අදින්න.

386. ආදානය කරනු ලබන එකෙහි ඇති ස්ඵර අකුරු / **Vowel Letters** ප්‍රතිදානයට අදාළ සිසුන් කිහිප දෙනෙකු විසින් ලියන ලද කේතයන් පහත දැක්වේ. ඒවා අතරින් නිවැරදි වන්නේ,

- i) 

```
A = input ('Enter a Word: ')
for k in range ( len(A) ):
    g = a[ k ]
    if g == 'a' or g == 'e' or g == 'i' or g == 'o' or g == 'u' :
        print(g)
```
- ii) 

```
A = input ('Enter a Word: ')
for k in range ( len(A) ):
    g = a( k )
    if g == 'a' or g == 'e' or g == 'i' or g == 'o' or g == 'u' :
        print(g)
```
- iii) 

```
A = input ('Enter a Word: ')
for k in range ( len(A) ):
    g = a[ k ].lower
    if g == 'a' or g == 'e' or g == 'i' or g == 'o' or g == 'u' :
        print(g)
```
- iv) 

```
A = input ('Enter a Word: ')
for k in range ( len(A) ):
    g = a[ k ].lower
    if g == 'a' or g == 'e' or g == 'i' or g == 'o' or g == 'u' :
        print(g)
```

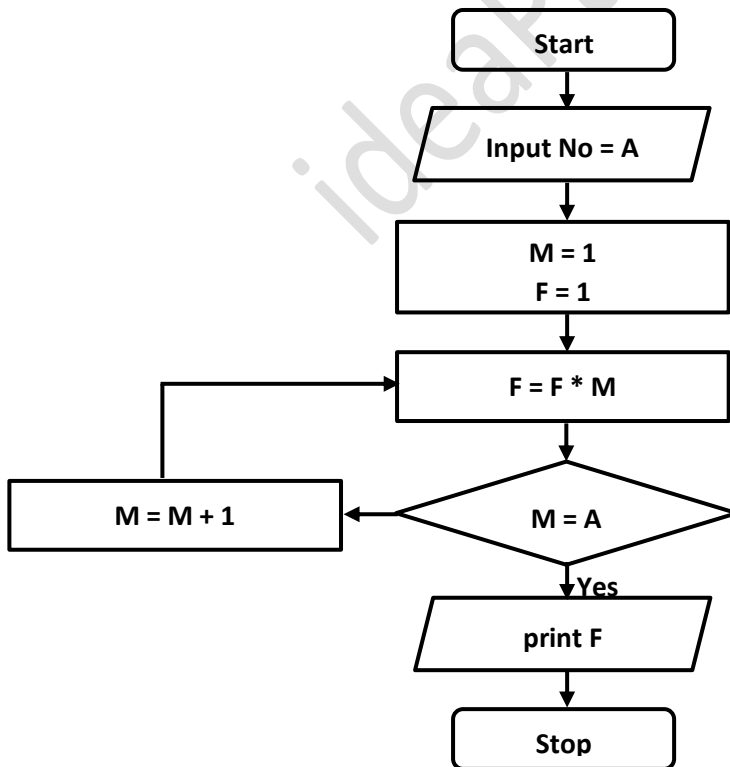
v) ඉහත සියල්ලම වැරදි. **Syntax Errors** පවතී.

387. පහත කේත කොටස සලකන්න.

```
A = [3]
B = [4]
print (A+B)
```

- i) [34]                      ii) [3, 4]                      iii) [7]                      iv) 7                      v) [3, 4, 7]

>>> පහත **Flow Chart** / ගැලීම සටහන සලකා 388 හා 389 ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.



388. ඉහත ඇල්ගොරිතමයට / **[Algorithm] 6** ලබාදුන් විට ලූපය / **Loop** කීවරක ක්‍රියාත්මක වේද,

- i) 2                      ii) 4                      iii) 5                      iv) 6                      v) 8

**389.** ඉහත ඇල්ගොරිතමය 4 ලබා දුන් විට ප්‍රතිදානය / output වන්නේ,

- i) 2
- ii) 4
- iii) 24
- iv) 22
- v) 76

**390.** පහත පයිතන් කේතයේ පවතින දෝෂය වන්නේ,

```
P = int(input("Enter Value : "))
X = 0
while X <= 5 :
    print (P)
X = X + 1
```

- i) **Syntax Error**
- ii) **Logic Error**
- iii) **Semantic Error**
- iv) **All**
- v) **No errors in the Code**

**391.** පරිගණක වැඩසටහනක් ලිවීමට පෙර ඇල්ගොරිතමයක් නිර්මාණය කිරීම නුවණට හුරු කරුණකි. මෙම ප්‍රකාශයට හේතුව වන්නේ,

- i) ඇල්ගොරිතමයක් ලිවීම කේතනයේදී අනිවාර්යය වේ.
- ii) ඇල්ගොරිතමයක් නිර්මාණයට පෙර කේත කොටස අවසන් කළ සිටිය යුතුය.
- iii) ඇල්ගොරිතමයක් නිර්මාණය කර පසුව කේතනය තුළින් කේතනයට වැය වන කාලය අඩු කර ගත හැකිය.
- iv) කේතනය හා ඇල්ගොරිතමය සමාන්තරව නිර්මාණය කරගෙන යාම මගින් කාලය ඉතිරි කරගත හැකිය.
- v) කේතනය හා ඇල්ගොරිතම නිර්මාණය අතර කිසිදු වෙනසක් නැත.

**392.** පහත කේතයේ ප්‍රතිදානය / output වන්නේ,

```
R = 'ideaPackk'
print('Max Letter : ' + min(R))
```

- i) space\*
- ii) P
- iii) a
- iv) k
- v) i

>>>. සංඛ්‍යාවක් ආදානය කළ පසුව එම අගය ආසන්න 10ට වැටීම සඳහා අදහස් කරමින් සකසන ලද පයිතන් කේතයක් පහත වේ. ඒ ඇසුරෙන් 393 සිට 395 දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
a = input('Enter : ')
b = int(a[-1])
if b >= 5 :
    print(int(a)-b)
else :
    print(int(a)+(10-b))
```

**393.** ඉහත කේතයට ආදානය ලෙස 56 ලබා දුන් විට ප්‍රතිදානය වන්නේ,

- i) 40
- ii) 60
- iii) 55
- iv) 100
- v) 50

**394.** ඉහත කේතයේ නිසි ආකාරව වැටීම සිදු නොවන බව සිසුවෙකු පවසයි. එය නිවැරදි බව පසුව අනාවරණය විය. එය නිවැරදි කිරීමට නම්,

- i) **b >= 5 ය b <= 5** ලෙස වෙනස් කිරීම
- ii) **4** ජේලිය හා අවසන් කේත ජේලිය එකිනෙක හුවමාරු කිරීම.
- iii) **int(a[-1]) ය int(a[1])** ලෙස වෙනස් කිරීම
- iv) **b >= 5 ය b < 5** ලෙස වෙනස් කිරීම
- v) **input('Enter :') ය int(input('Enter :'))** ලෙස වෙනස් කිරීම

**395.** ආදානය / input කරනු ලබන අගය a හි ගබඩා / save වන්නේ,

- i) Char
- ii) String
- iii) Double
- iv) Integer
- v) Float

396. `print (int (input ('Enter a Value : ')) ** 2)` මෙම කේතය සඳහා 10 ආදානය / input කළ විට ප්‍රතිදානය / output වන්නේ,

- i) 100.00
- ii) 1000
- iii) 100
- iv) Nothing
- v) Error

397. `print( int ( input ('Enter Val 2 : ')) + int ( input ('Enter Val 1') ) + 2)` මෙම කේතය සම්බන්ධයෙන් ඇති නිවැරදි ප්‍රකාශය / ප්‍රකාශ වන්නේ,

- a. ආදාන / inputs දෙකක් ලබා දීමට ඉඩ සලසා දේ.
- b. පළමු ආදානය / first input ලබා ගැනීමට කේතය විසින් Enter Val 2 Display කරවයි.
- c. 12 හා 44 ආදානය කළ විට 56 ප්‍රතිදානය වේ.

- i) a පමණි.
- ii) c පමණි.
- iii) a , b පමණි.
- iv) a , c පමණි.
- v) සියල්ලම.

398. ඉහත 397 කේතය සඳහා පිළිවෙලින් 10, 33 ආදානය කරන ලදී. එවිට ප්‍රතිදානය වන්නේ,

- i) 20.0
- ii) 21.5
- iii) 21
- iv) 26.5
- v) 45

399. ක්‍රමලේඛණයේදී භාවිතාවන Comments සම්බන්ධයෙන් ඇති නිවැරදි ප්‍රකාශය / ප්‍රකාශ වන්නේ,

- i) කේතයේ ආරම්භයේදී පමණක් Comment එකක් යෙදීම නුවණට හුරු කරුණකි.
- ii) කේතයේ සෑම ජේළියකම Comment එකක් යෙදීම හොඳ ක්‍රමලේඛණ පුරුද්දකි.
- iii) අවශ්‍ය තැන් වල පමණක් Comments යෙදීම වඩා නිවැරදි ක්‍රමයකි.
- iv) සමහර ක්‍රමලේඛණ භාෂාවල අකුරෙන් C ආරම්භ Comment කරයි.
- v) ඉහත iii හා iv යන දෙකම නිවැරදි.

400. හොඳ ක්‍රමලේඛකයෙකුගේ ලක්ෂණ / ලක්ෂණයක් වන්නේ,

- i) කේතනයේදී වැඩි වේලාවක් ලබා ගෙන නිවැරදිව එක් වරකින්ම නිවැරදි කේතය ලිවීම.
- ii) කේත කොටස් මතකයේ තබා ගනිමින් කේතය ලිවීම.
- iii) ඉතා ඉක්මනින් කේතය ලිවීම.
- iv) අදි වේගීව යතුරු ලියන හැකියාව.
- v) අඩු කාලයකින් තම Project එක නිවැරදිව අවසන් කිරීම.

401. පරිගණකයට තේරුම් ගත හැකි භාෂාව හඳුන්වන නාමය වන්නේ,

- i) මූල කේතය
- ii) අමු කේතය
- iii) ප්‍රභව කේතය
- iv) යාන්ත්‍රික කේතය
- v) ඉහත සියල්ලම

402. ක්‍රමලේඛ භාෂා සම්බන්ධයෙන් පහත සඳහන් කුමන ප්‍රකාශ සත්‍යවේද ?

- i) පළමු පරම්පරාවේ භාෂාවන් හී ක්‍රමලේඛණ ලිවීම ඉතා පහසු විය.
- ii) පළමු පරම්පරාවේ භාෂාවන් හී දෝෂ නිදොස් කිරීම ඉතා පහසුය.
- iii) තුන්වන සහ හතරවැනි පරම්පරාවේ භාෂා මගින් ක්‍රමලේඛණයක් සැකසීමට දෘඩාංග පිළිබඳ හොඳ දැනුමක් තිබිය යුතුය.
- iv) පළමු පරම්පරාවේ භාෂා යන්ත්‍ර මත යැපේ.
- v) ඉහත සියල්ලම නිවැරදි.

403. පහත ඒවා අතරින් පරිගණක භාෂාවක් නොවන්නේ,

- i) Java
- ii) A.I.Langu
- iii) C
- iv) C#
- v) Prolog

404. දර්පණ තම නව නිර්මාණයක් සඳහා මෘදුකාංගයක් නිපදවමින් සිටියා. එහි දෘඩාංග කොටසක් ද පවතී. එමනිසා දෘඩාංග කොටසේ වැඩ නිසා මෘදුකාංගය නිපදවීම සතියකට කල් දමන ලදී. පසුව දෘඩාංග කොටස නිම කෙරු දර්පණ නැවත සතියකට පසු මෘදුකාංගය නිපදවීම සඳහා කේතනයේ යෙදීමට සැරසිය. නමුත් ඇයගේ කේතය ඇයටම තේරුම් ගැනීමට අපහසු විය. එනම් කේතයේ එක් එක් කොටසෙන් සිදුවන කාර්ය ඇයට අමතක වී තිබිණි. මේ ගැටලුව නැවත සිදුවීම වළකා ගැනීමට දර්පණට දිය හැකි වඩාත් සුදුසුම පිළිතුර වන්නේ,

- i) සම්පූර්ණම කේතයම අවසන් වනතුරු නොකඩවා සිදුකර අවසන් කිරීම.
- ii) කේතයේ, comment අවශ්‍ය පරිදි යෙදීම.

- iii) කේතයේ සිදුවන කායියන් කඩදාසි කොළය ලියා තබා ගැනීම.
- iv) කේතය මතකයේ තබා ගැනීම.
- v) ඉහත සියල්ලම ප්‍රායෝගිකව සිදු කිරීමට හැකි සුදුසු ක්‍රම වේ.

405. නිවසක් නිපද වීමට පෙර සැලසුමක් නිර්මාණය කරනවා මෙන් කේතයක් ලිවීමට පෙර,

- i) ගැලිම් සටහනක් ඇදීම. **Flow Chart**
- ii) කුඩා ප්‍රමාණයේ නිවැරදි කේතයක් පළමුව නිර්මාණය කිරීම. / **MallCoding**
- iii) ව්‍යාජ කේතයක් ලිවීම. **Pseodu Code**
- iv) ඉහත i හා iii දෙකම නිවැරදි.
- v) සියල්ලම නිවැරදි.

406. බොහෝ සිසුන් තනිව කේතනය කිරීමට වඩා අන්තර්ජාලයෙන් කේත බාගත කර එමගින් තම එක නිර්මාණය කිරීමට පුරුදුවී ඇත. මෙහි ප්‍රතිපලයක් ලෙස,

- i) ඉතා හොඳ ක්‍රමලේඛකයන් විශාල පිරිසක් බිහිවේ.
- ii) අන්තර්ජාලයට කේත සපයන ක්‍රමලේඛකයන් ප්‍රමාණය ඉහළ යයි.
- iii) සිසුන්ට ලේසියෙන් ක්‍රමලේඛකයෙකු වීමට අවකාශය උදාවේ.
- iv) හොඳ ක්‍රමලේඛකයන් බිහිවීම අඩුවේ.
- v) කේතනයට වැයවන වේලාව ඉතිරි වන නිසා එම වේලාවේදී ක්‍රමලේඛකණය පිළිබඳව හැදෑරීම කළ හැකිය.

407. ක්‍රමලේඛණය සම්බන්ධයෙන් ඇති පහත ප්‍රකාශ වලින් නිවැරදි ප්‍රකාශය / ප්‍රකාශ වන්නේ,

- i) කේතනය / **Progrmming** මගින් සෑම විටම මෘදුකාංග / **Software** නිපදවනු ලබයි.
- ii) **Smart Phone** මගින් ක්‍රමලේඛණය කළ නොහැකිය.
- iii) ක්‍රමලේඛණ ධාවනය කළ හැක්කේ / **Micro Processor** තුළ පමණි.
- iv) සෑම විටම ක්‍රමලේඛණය සිදු කිරීමට යතුරු පුවරු / **Keyboard** භාවිතා කිරීමට සිදුවේ.
- v) ඉහත සියල්ලම වැරදි.

>>> පහත **Code** / කේතය සලකා **408** හා **409** ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```

B = 0
while B < 5:
    F1 = open( 'f1.txt' , 'a')
    F2 = open( 'f2.txt' , 'a')
    A = int (input ( 'Enter a Value : '))
    if A % 2 == 0 :
        F1.write(str(A)+ '\n')
    else :
        F2.write(str(A)+ '\n')
    B = B + 1
F1.close() ; F2.close()
    
```

408. ඉහත කේතයේ කායිය වන්නේ,

- i) ආදානය කරන අගයේ ප්‍රතිශය හා භාග අගය වෙන වෙන දෙකක ගබඩා කරවීම.
- ii) ඕත්තේ හා ඉරට්ටේ ලෙස අගයන් වෙන්කර **File** දෙකක ගබඩා කිරීම.
- iii) දෙකෙන් බෙදෙන හා නොබෙදෙන ලෙස අගයන් වෙන්කර **File** දෙකක ගබඩා කිරීම.
- iv) අහඹු ලෙස අගයන් ගබඩා කරවීම.
- v) ඉහත කිසිවක් නොවේ.

409. ඉහත කේතයට අනුව ආදාන කොපමණ ප්‍රමාණයක් ලබා ගනීද,

- i) 2
- ii) 3
- iii) 20
- iv) 5
- v) කිව නොහැකිය

**410. while True** ලූපය සම්බන්ධයෙන් පහත ප්‍රකාශ සලකා නිවැරදි ප්‍රකාශය තෝරන්න.

- i) ලූපය නැවැත්වීමට නොහැකිය
- ii) ලූපය 10 වරක් හෝ ඊට අඩුවෙන් ගමන් කරන අතර කේතය දිගටම ගමන් කරයි.
- iii) **break** මගින් ලූපය නතර කිරීමේ හැකියාව ඇත.
- iv) **continue** මගින් ලූපය තාවකාලිකව නතර කළ හැකිය.
- v) ඉහත සියල්ලම නිවැරදි.

>>> පහත **Code / කේතය සලකා 411 සිට 413** ප්‍රශ්න සඳහා පිළිතුරු සපයන්න. පහත දක්වා ඇත්තේ පයිතන් ගොනු භාවිතය සම්බන්ධ ප්‍රශ්නයකි. පහත දැක්වෙන්නේ එම ගොනුව සහ පයිතන් කේතයකි.

```

a - Notepad
File Edit Format View Help
10
11
32
01
77
90
    
```

```

f1 = open('a.txt','r')
a = f1.readline()
while a:
    a = f1.readline() . strip()
    print(a , end = ',')
    
```

**411.** ඉහත කේතයේ ප්‍රතිදානය වන්නේ,

- i) 11,32,01,77,90,
- ii) 10,11,32,01,77,90 පහළට
- iii) No Outputs
- iv) Error
- v) 10,11 and Error

**412. a** නම් වූ **Variable** එකෙහි ගබඩා වන **Variable Type** වන්නේ,

- i) int
- ii) string
- iii) float
- iv) double
- v) char

**413.** ඉහත ගොනුවේ ගබඩා වී ඇති අගයන් සියල්ලගේම එකතුව ප්‍රතිදානය සඳහා ඉහත කේතයේ කළ යුතුය. නිවැරදි කේතය වන්නේ,

```

i) f1 = open('a.txt','r')
a = f1.readline()
while a:
    a = f1.readline()
    print (a, end = ',')
    Total = Total + a
    
```

```

ii) f1 = open('a.txt','r')
a = f1.readline()
while a:
    a = f1.readline()
    print (a, end = ',')
    Total = Total + a
    
```

```

iii) f1 = open('a.txt','r')
a = f1.readlines()
while a:
    a = f1.readline()
    print (a, end = ',')
    Total = Total + a
    
```

```

iv) f1 = open('a.txt','r')
a = f1.readline()
while a:
    Total = Total + a
    a = f1.readline()
    print (a, end = ',')
    
```

- v) ඉහත සියල්ලම වැරදි

>>> පහත **Code** / කේතය සලකා **414** සිට **416** දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
A = []
for i in range(6):
    P = int(input("Enter a No : "))
    A.append(P)
    A[0] = max(A)
    A[-1] = min(A)
print(A)
```

**414.** පහත ප්‍රකාශ අතරින්, ඉහත කේතය සම්බන්ධයෙන් ඇති නිවැරදි ප්‍රකාශය වන්නේ,

- ආදානය කරනු ලබන අගයන් අවරෝහණ පිළිවෙලට සකස් කිරීම.
- ආදානය කරනු ලබන අගයන් ආරෝහණ පිළිවෙලට සකස් කිරීම.
- ආදානය කරනු ලබන අගයන් පිළිවෙලකට සැකසීම.
- විශේෂිතවූ **task** එකක් නැත.
- ආදානය කරන අගයන් 4 ගෙන එකට එකතු කර ප්‍රතිදානය කරවීම.

**415.** 1, 5, 7, 3, 5, 8 යන අගයන් පිළිවෙලකට ආදානය කළ විට ලැබෙන ප්‍රතිදානය වන්නේ,

- [7, 5, 5, 3, 3]
- [1, 3, 5, 5, 7]
- [7, 5, 5, 3, 3, 8]
- [8, 5, 5, 3, 3, 3]
- [29]

**416.** ඉහත කේතයට අදාළ ගැලීම් සටහන අදින්න.

>>> පහත **Code** / කේතය සලකා **417** සිට **421** දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න. පහත දක්වා ඇත්තේ නම් වූ ගොනුවේ දත්ත ගබඩා කර ඇති ආකාරයයි. එක් එක් ළමුන් ගණනයට ලබා ගත් ලකුණු පහත වේ.

```
a - Notepad
File Edit Format View Help
Bimsara 66
Kalana 78
Darpani 45
Dinethi 88
Kavindu 78
Bimsara 90
```



417. ළමයෙකුගේ නම ඇතුළත් කළ විට ඔහුගේ හෝ ඇයගේ ලකුණු ප්‍රතිදානය කර ගැනීමට අදාළ නිවැරදි කේතයක් පහත වේ. එහි කේත ජේලී / Code Line දෙකක් මැකී ඇත. එහි A ට අදාළ කේත Line එක වන්නේ,

```

a = input('Enter Name for Marks : ')
..... ← A
c = f1.readline()
while c :
    ..... ← B
    if a == b[0] :
        print(b[1])
        break ← C
    c = f1.readline()
f1.close()
    
```

- i) f = open('notepad.txt', 'r')
- ii) f = open('a.txt', 'r+')
- iii) open = f('a','r')
- iv) f1 = open('a.txt', 'r')
- v) f1 = open('b.txt', 'r')

418. එහි B ට අදාළ කේත එක වන්නේ,

- i) b = f1.readline()
- ii) b = f1.read()
- iii) b = c.split(' ')
- iv) b = readline().split().strip(',')
- v) b = readline().strip().split(',')

419. අමල් ඉහත කෙතයේ මැකුණු කේත ජේලී නිවැරදි කර, කේතය නිවැරදිව ක්‍රියාකරන ලෙස සකස් කරගෙන ඇත. ඉන් පසු 'Bimsara' ලෙස ආදානය / input කරන ලදී. එවිට ප්‍රතිදානය / output වන්නේ,

- i) 66
- ii) 90
- iii) 66, 90
- iv) Error
- v) 66  
90

420. darpani ලෙස ආදානය කරන ලදී. එවිට ප්‍රතිදානය / output වන්නේ,

- i) 60
- ii) 45
- iii) '45'
- iv) No output
- v) Error

421. File එකක් නිර්මාණය කිරීමට අදාළ නිවැරදි කේත ජේලීය නොවන්නේ,

- i) f = open('n.txt', 'w')
- ii) f = open('a.txt', 'w+')
- iii) f = open ('a',' w')
- iv) f1 = open('a.txt', 'r+')
- v) f1 = open('b.txt', 'a')

>>> පංතියක සිටින ළමුන් ගණිතය සඳහා ලබා ගත් ලකුණු Database එකකට ඇතුළත් කිරීමට අවශ්‍ය වී ඇත. දත්ත ගබඩා කරන්න text file එකක් තුළ වන අතර 417 සිට 421 දක්වා ප්‍රශ්න සඳහා අදාළ වූ text File එකේ ආකාරයටමය. ඒ සඳහා සිසුවෙකු සැකසූ පයිතන් කේතය පහත වේ. නමුත් එහි කේත ජේලී දෙකක් මැකී ඇත. ළමුන් 20 දෙනෙකු පංතියේ සිටියි. 422 හා 423 ප්‍රශ්න සඳහා ඒ ඇසුරෙන් පිළිතුරු දෙන්න.

```

for l in range(20) :
    A = input('Enter Your Name : ')
    B = ..... ← X
    F1 = open('a.txt','a')
    C = str(A) + ' ' + Str(B)
    F1.write(C)
..... ← Y
    
```

422. ඉහත කේතයේ X හිස්තැනට සුදුසුම කේත ඡේදය වන්නේ,

- i) `input('Enter Your Marks :')`
- ii) `int(input('Enter Your Marks :'))`
- iii) `float(input('Enter Your Marks :'))`
- iv) `long(input('Enter Your Marks :'))`
- v) ඉහත සියල්ලම යෙදිය හැකිය.

423. ඉහත කේතයේ Y හිස්තැනට සුදුසුම කේත ඡේදය වන්නේ,

- i) `F1 = close()`
- ii) `end while`
- iii) `close(F1)`
- iv) `F1.close()`
- v) `close('b.txt', 'a')`

424. පහත කේතයේ ප්‍රතිදානය වන්නේ,

```
A = [1, 2, 3]
A.pop(2)          #This pop function doesn't support to list that have string type values
print(A)
```

- i) `[1, 3]`
- ii) `[1, 2]`
- iii) `[2, 3]`
- iv) `[1, 2, 3, 3]`
- v) Error

425. පහත කේතයේ ප්‍රතිදානය වන්නේ,

```
A = [1, 2, 3]
A.remove(2)      #This remove function doesn't support to list that have string type values
print(A)
```

- i) `[1, 3]`
- ii) `[1, 2]`
- iii) `[2, 3]`
- iv) `[1, 2, 3, 3]`
- v) Error

>>> පහත Code / කේතය සලකා 426 සිට 428 දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න. පහත කේතය වෙනම ආදානය කරන වචන ආරෝහණ පිළිවෙලට ප්‍රතිදානය එහි කාර්යයි.(මෙම රූපයේ ආකාරයට).

```
a = [ ]
for i in range(5):
    b = input('Enter a word : ')
    a.append(b)

for i in range(5):
    .....
    del a[a.index(max(a)) ]
```

1<sup>st</sup> for Loop (points to the first for loop)

2<sup>nd</sup> For Loop (points to the second for loop)

Deleted Code Line (points to the del statement)

426. ඉහත කේතයේ පළමු ෆෝර් ලූපයේ / 1<sup>st</sup> for Loop කාර්යය වන්නේ,

- i) විශාලම දත්තය List එකේ පළමු තැන සිටින සේ ඇතුළත් කිරීම.
- ii) දත්ත 4ක් ආදානය කර ගෙන එය එකක List ගබඩා කිරීම.
- iii) දත්ත ආදානය හා එවා List එකක් තුළ ගබඩා කර ගැනීම.
- iv) දත්ත ආදානය හා එවා පිළිවෙලකට සැකසීම.
- v) ඉහත කිසිවක් නොවේ.

427. Deleted Code Line සඳහා වඩාත්ම සුදුසුම කේත ඡේදය වන්නේ,

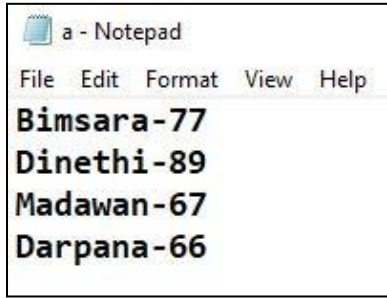
- i) `print(min(a))`
- ii) `print(maximum(a))`
- iii) `print(max(a))`
- iv) Nothing
- v) i or ii

428. 427 දී නිවැරදි ව සකසන ලද කේතය සඳහා 'dar' 'tar' 'sar' 'ar' 'far' ලෙස ආදාන ඇතුළත් කළ විට ප්‍රතිදානය විය හැක්කේ,

- i) tar,sar,far,dar,ar ලෙස පහළට
- ii) sar,far,dar,tar,ar ලෙස පහළට
- iii) tar,sar,far,ar,dar ලෙස පහළට
- iv) ar,sar,far,dar,tar ලෙස පහළට
- v) Error

>>> පහත Code / කේතය සලකා 429 හා 430 ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
F = open('a.txt','r')
Read = F.readline()
e = 0
while Read:
    d = Read.strip().split('-')
    Read = F.readline()
    e = e + int(d[1])
print(e)
```



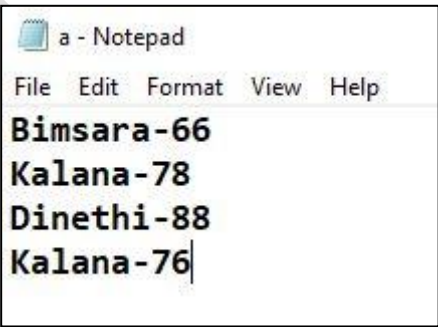
429. ඉහත කේතයේ ප්‍රතිදානය / output වන්නේ,  
 i) -20                      ii) 89                      iii) 77                      iv) 299                      v) Error

430. ඉහත කේතය නිවැරදිව ක්‍රියාත්මක වුවද යම් කිසි අඩුවක් පවතින බව මනිසු පවසයි.එයට හේතුව එනම් අඩුව වන්නේ,  
 i) a.txt file එක open කර නැත.  
 ii) a.txt file එක close කර නැත.  
 iii) F variable එකේ data type එක ලබා දී නැත.  
 iv) ඉහත සියල්ලම.  
 v) ඉහත කිසිවක් නොවේ.

>>> පහත Code / කේතය සලකා 431 සිට 434 දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.



A



B

431. A ආකාරයට දිස් වූ එක B ආකාරයට වන ලෙස දත්ත ජේළි 2 ක් ඇතිලත් කිරීමට සැකසූ කේතයක් පහත වේ.එහි A හිස්තැනට සුදුසු කේත ජේළිය වන්නේ,

```
A = open ('a.txt', 'a')
A.write ('\n')
..... ← A
A.write ('Kalana-76')
A.close ()
```

- i) A.write('\n')                      ii) A.write('Dinethi-88' + '\n')                      iii) A.write('Dinethi-88/n')  
 iv) A.write('Dinethi') ; A.write('-88') ; A.write('\n')                      v) ii and iv

432. ඉහත ප්‍රශ්නයේ සඳහන් a.txt හි ගබඩා කර ඇති ආකාරයටම මූල පන්තියේම ළමුන් ගේ ලකුණු ගබඩා කර ඇත. ළමුන් ගණන නිශ්චිත නැත. පංතියේ ළමුන් ගේ ලකුණු සාමාන්‍ය ලබා ගැනීමට (ළමුන්ගේ ලකුණු වල මූල එකතුව / ළමුන් ගණන) සකස් කරන ලද කේත වලින් නිවැරදිම කේතය වන්නේ,

```
i) F = open('a.txt','r')
   G = F.readline()
   Count = 0
   Total = 0
   while G:
       T = G.strip().split('-')
       Total = Total + int(T[1])
       Count += 1
       G = F.readline()
   print('Average = ', Count / Total)
```

```
ii) F = open('a.txt','r')
     G = F.readline()
     Count = 0
     Total = 0
     while G:
         T = G.strip().split('-')
         Total = Total + int(T[1])
         Count += 1
         G = F.readline()
     print('Average = ', Total / Count)
```

```
iii) F = open('a.txt','r')
      G = F.readline()
      Count = 1
      Total = 0
      while G:
          T = G.strip().split('-')
          Total = Total + int(T[1])
          Count += 1
          G = F.readline()
      print('Average = ', Total / Count)
```

```
iv) F = open('a.txt','r')
     G = F.readline()
     Count = 0
     Total = 0
     while G:
         T = G.strip().split('-')
         Total = Total + T[1]
         Count += 1
         G = F.readline()
     print('Average = ', Total / Count)
```

iv) ඉහත කිසිවක් නොවේ.

433. පහත කේතයේ ප්‍රතිදානය ලෙස B = ['Bimsara', 'Kalana', 'Dinethi', 'Kalana'] මේ පරිදි ගොනුවේ දත්ත වල නම් List එකක ගබඩා කරවීමට අවශ්‍ය වී ඇත. ඒ සඳහා සැකසූ කේතය පහත වේ. එහි A හිස්තැනට සුදුසුම කේත පේළිය වන්නේ, [425 ඡේදනයේදී භාවිතා කළ a.txt ගොනුවම මෙහිදී යොදා ගනී.]

```
B = []
F = open('a.txt','r+')
R = F.readline()
while R:
    .....
    B.append(T[0])
    R = F.readline()
print(B)
F.close()
```

A space

- i) `T = F.strip().split()`                      ii) `B = T.strip().split()`                      iii) `T = R.strip().split('-')`  
 iv) `B = R.strip().split()`                      v) Error

**434.** ඉහත ප්‍රශ්නයේ ලබා ගත් නාමයන් List එක වෙනුවට ලකුණු සහිත List එකක් ලබා ගැනීමට 433 හි සැකසූ කේතයේම සිදු කළ යුතු වෙනස් කම වන්නේ,

- i) `B.append(T[0])`                      ➔                      `B.append(T[1])`  
 ii) `R = F.readline()`                      ➔                      `F = R.readline()`  
 iii) `B.append(T[0])`                      ➔                      `B.append(T[-1])`  
 iv) `print(B)`                      ➔                      `print(T)`  
 v) i or iii

**435.** පහත කේතයේ ප්‍රතිදානය වන්නේ,

```
A = ['Danush','Akash','Cristiana','Banuka']
Listlen = len(A)
for i in range(Listlen - 1) :
    for h in range(i+1, Listlen) :
        if A[ i ] > A[ h ] :
            A[ i ], A[ h ] = A[ h ], A[ i ]
print(A)
```

- i) `A = ['Danush','Akash','Cristiana','Banuka']`                      ii) `A = ['Danush','Cristiana','Banuka','Akash']`  
 iii) `A = ['A','B','C','D']`                      iv) `A = ['Akash','Banuka','Cristiana','Danush']`  
 v) Error

**436.** පහත කේතයේ ප්‍රතිදානය වන්නේ,

```
A = ['Danush','Akash','Cristiana','Banuka']
Listlen = len(A)
for i in range(Listlen - 1) :
    for h in range(i+1, Listlen) :
        if A[ i ] < A[ h ] :
            A[ i ], A[ h ] = A[ h ], A[ i ]
print(A)
```

- i) `A = ['Danush','Akash','Cristiana','Banuka']`                      ii) `A = ['Danush','Cristiana','Banuka','Akash']`  
 iii) `A = ['A','B','C','D']`                      iv) `A = ['Akash','Banuka','Cristiana','Danush']`  
 v) Error

**437.** පහත කේතය ධාවනය කිරීමෙන් පසු සිදුවන්නේ,

```
for i in range (3):
    F = open ('p.txt', 'w')
    F.write ('ideapacklk\n') ; F.close ()
```

- i) `ideapacklk ideapacklk` ලෙස `p.txt` ගොනුව තුළ ගබඩා වේ.  
 ii) `ideapacklk` ලෙස 3වරක් පහළට ලෙස, `p.txt` ගොනුව හී ගබඩා වේ.  
 iii) `Ideapacklk` ලෙස එක් වරක් පමණක් `p.txt` ගොනුව හී ගබඩා වේ.  
 iv) `Ideapacklk\n ideapacklk\n ideapacklk\n` ලෙස `p.txt` ගොනුව තුළ ගබඩා වේ.  
 v) Error or Nothing write in p.txt file

438. පහත කේතය ධාවනය කිරීමෙන් පසු සිදුවන්නේ,

```
for i in range (3):
```

```
    F = open ('p.txt', 'a')
```

```
    F.write ('ideapacklk\n')
```

```
    F.close ()
```

- `ideapacklk ideapacklk ideapacklk` ලෙස `p.txt` ගොනුව තුළ ගබඩා වේ.
- `ideapacklk` ලෙස 3වරක් පහළට ලෙස, `p.txt` ගොනුව හී ගබඩා වේ.
- `ideapacklk` ලෙස එක් වරක් පමණක් `p.txt` ගොනුව හී ගබඩා වේ.
- `ideapacklk\n ideapacklk\n ideapacklk\n` ලෙස `p.txt` ගොනුව තුළ ගබඩා වේ.
- Error or Nothing write in `p.txt` file

439. පහත කේතය ධාවනය කිරීමෙන් පසු සිදුවන්නේ,

```
for i in range (3):
```

```
    F = open ('p.txt', 'a')
```

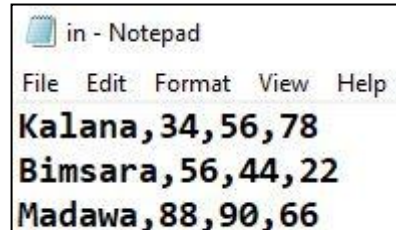
```
    F.write ('srilanka ')
```

```
F.close ()
```

- `srilanka srilanka srilanka` ලෙස `p.txt` ගොනුව තුළ ගබඩා වේ.
- `srilanka` ලෙස 3වරක් පහළට ලෙස, `p.txt` ගොනුව හී ගබඩා වේ.
- `srilanka` ලෙස එක් වරක් පමණක් `p.txt` ගොනුව හී ගබඩා වේ.
- `srilanka\n srilanka\n srilanka\n` ලෙස `p.txt` ගොනුව තුළ ගබඩා වේ.
- Error or Nothing write in `p.txt` file

>>> පහත කේතයේ පයිතන් හී ගොනු භාවිතය සම්බන්ධ කේතයකි. ගොනු දෙකක් පවතී. එකක් `in.txt` සහ අනෙක් `out.txt` වේ. `in.txt` ගොනුව පහත ආකාරයෙන් දිස් විය. එහි සඳහන් වන්නේ පංතියේ ළමුන් තුන් දෙනාගේ සිංහල, බුදු දහම, ගණිතය යන විෂයන්ට ළමුන් ලබා ගත් ලකුණු වේ. ඒ ඇසුරෙන් 440, 441 ප්‍රශ්න වලට පිළිතුරු ලියන්න.

```
inp = open ('in.txt', 'r')
out = open ('out.txt', 'w')
line = inp.readline()
while line :
    d = line.strip().split(',')
    total = int(d[1]) + int(d[2]) + int(d[3])
    out.write( '{ }, { } \n' . format(d[0], total) )
    line = inp.readline()
out.close(); inp.close() ← #last line
```



```
in - Notepad
File Edit Format View Help
Kalana, 34, 56, 78
Bimsara, 56, 44, 22
Madawa, 88, 90, 66
```

440. කේතය එක් වරක් ධාවනයෙන් පසු `out.txt` ගොනුව දිස් වෙන ආකාරය වන්නේ,

- සිසුවාගෙ නම හා ඔහු විෂයන්ට ලබාගත් ලකුණු වල සාමාන්‍ය අගය.
- සිසුවා ලබා ගත් ලකුණු සාමාන්‍ය අගය.
- සිසුවාගෙ නම හා ඔහු විෂයන්ට ලබාගත් ලකුණු වල මුලු එකතුව.
- සිසුවාගේ නම හා ගණිතයට ලබා ගත් නම.
- ඉහත කිසිවක් නොවේ.

441. අමල් විසින් පෙර අවස්ථාවේදී `out.txt` හී ලියවන ලද දත්ත සියල්ලම ගොනුවෙන් ඉවත් කරන ලදී. දැන් `out.txt` යනු නිස් ගොනුවකි. `in.txt` හී පෙර තිබූ දත්ත වෙනසකින් තොරව එලෙසම පවතී. ඉන් පසු අමල් විසින් අත් හදා බැලීමක් ලෙස කේතයේ අවසන් පේළිය / last line මකා දමා කේතය පෙර ලෙසම ධාවනය කරන ලදී. අනතුරුව ඔහු පෙර ලෙස ගොනු විවෘත කර බැලූ විට,

- දත්ත පෙර පරිදීම දිස් විය.
- කිසිදු දත්තයක් නොමැති විය.

- iii) නිශ්චිතව කිව නොහැකිය.
- iv) **Error** පණිවිඩ දිස් වීමට පටන් ගැනුණේය.
- v) දත්ත අඩුවෙන් ගබඩා වී පැවතීය.

>>> පහත කේතය ඇසුරෙන් **442 , 443, 444** හා **445** ප්‍රශ්න සඳහා පිළිතුරු සපයන්න. පහත දැක්වෙන්නේ සිසුවකු පයින් භාවිතා කර සැකසූ ඉතා සරල **Game** එකකි. [**Game Name: Guess the X if U can**]

```
X,marks = 5,10
Y = int (input ('Enter plus Value: '))
while (X != Y and marks > 0):
    if ( X - Y < 4 and X - Y > -4 ) :
        print('Hot')
    else:
        print('Cool')
    Y = int (input ('Enter plus Value: '))
    marks = marks - 1

print('You Won!! Your Marks : ', marks)
```

- 442.** ඉහත කේතය සම්බන්ධයෙන් දැක්වෙන පහත ප්‍රකාශ අතරින් නිවැරදි ප්‍රකාශය / ප්‍රකාශ වන්නේ,
- a. 8 ඇතුළත් කළ විට **Hot** ලෙස **print** කරවයි.
  - b. X හි අගය අනුමාන කිරීමට අවස්ථා **10**ක් ලබා දේ.
  - c. අවස්ථා **2**ක් පමණක් භාවිතා කර X අගය නිවැරදිව අනුමාන කළ විට ලකුණු **8**ක් හිමි වේ.
- i) a පමණි.      ii) b පමණි.      iii) a, b පමණි.      iv) a, c පමණි.      v) කිසිවක් නොවේ.

**443.** දුලාන් **12, 13, 6, 7, 4, 3, 7, 5** ලෙස 8 වරක් උත්සහ කර තරග ජය ගත්හ. එවිට ඔහුට හිමිවන ලකුණු ප්‍රමාණය වන්නේ,

- i) 3                      ii) 4                      iii) 2                      iv) 8                      v) 7

**444.** ඉහත **443** හි ආදානම ලබා දුන් විට **Cool** ලෙස හා **Hot** ලෙස ප්‍රතිදානය කරන වාර ගණන පිළිවලින්,

- i) 3, 5                      ii) 2, 6                      iii) 3, 6                      iv) 5, 2                      v) 2, 5

**445.** ඉහත කේතයට / code අදාළ ගැලීම් සටහනක් / **Flow Chart** අඳින්න.

**446.** පහත කේතයේ ප්‍රතිදානය / **output** වන්නේ,

```
def Bim ():
    print ('ideaPack')
    Bim ( )
Bim ()
```

- i) **ideaPack** ලෙස එක් වරක් ප්‍රතිදානය වේ.
- ii) **ideaPack** ලෙස දෙවරක් ප්‍රතිදානය වේ.
- iii) **ideaPack** ලෙස ප්‍රතිදානය වී දෝෂ පණිවිඩයක් දිස් කරයි.
- iv) **ideaPack** ලෙස නො නවත්වා ප්‍රතිදානය කරයි.
- v) ඉහත කිසිවක් නොවේ.

447. මෙම කේතය මගින් ලබුණු ගේ BMI අගය පරීක්ෂා කිරීමට යි.

[ BMI = ( Weight [kg] / (Height \* Height [m] ) ). මෙහි A නම් හිස්තැනට ගැලපෙන කේත ජේළිය වන්නේ,

```
N = input ( "Enter Your Name : " )
H = float (input( "Enter Your Height (m): " ))
W = float (input( "Enter Your Height (kg): " ))
BMI = .....
print(BMI)
```

#A sapce

- i)  $M * M / W$
- ii)  $M * 2 / W$
- iii)  $W / M * M$
- iv)  $W / H * 2$
- v)  $W / H * 2$

448. 448 දී ලබා ගත් නම හා ගණනය කර ගත් BMI අගය ලබා දුන් විට එම දත්ත පහත ආකාරයට ගොනුව තුළ ගබඩා කරවීමට ශ්‍රිතයක් / Function නිවැරදිව නිර්මාණය කර ඇත්තේ,



- i) `def Writing(n, bmi) :`  
`F = open('database.txt' , 'a')`  
`F.write('{} , {} \n' . format(n, bmi))`
- ii) `def Writing(n, bmi) :`  
`F = open('database.txt' , 'a')`  
`F.write('{} , {} \n' . format(N, BMI))`
- iii) `def Writing() :`  
`F = open('database.txt' , 'a')`  
`F.write('{} , {} \n' . format(N, BMI))`
- iv) `def Writing(N, BMI) :`  
`F = open('database.txt' , 'a')`  
`F.write('{} , {} \n' . format(n, BMI))`
- v) Nothing

>>> පහත කේතය ඇසුරෙන් 449, 450 හා 451 ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
i , S = 10, 0
for p in range(i , i, 5) :
    if p > 0 :
        S = S + 1
    else :
        S = S
S = i / S
print(S)
```

449. ඉහත කේතය ක්‍රියාත්මක කිරීමේදී දෝෂ පණිවිඩයක් දිස් විය එයට හේතුවන කේත ජේළිය මේ අතරින්, Error Message = ZeroDivisionError: division by zero

- i) 3 ජේළිය
- ii) 2 ජේළිය
- iii) 5 ජේළිය
- iv) 4 ජේළිය
- v) 6 ජේළිය

450. දෝෂ රහිත කේතයක් ලබා ගැනීමට ඉහත කේතයේ කළ හැකි පහත වෙනස් කම වන්නේ,

- i) `i , S = 10, 0` → `i , s = 10, 0` ලෙස
- ii) `for p in range(i , i, 5)` → `for p in range(-i , i, 5)` ලෙස
- iii) `else:` → `elif :` ලෙස
- iv) `S = S + 1` → `S = 1 + S` ලෙස
- v) `S = i / S` → `S = i*3 / S` ලෙස



**451. 450** දෝෂ රහිත කේතයක් ලබා ගැනීමට කළ වෙනස්කම පසුව ප්‍රතිදානය වන්නේ,  
 i) 1                      ii) 2                      iii) 5.0                      iv) 4                      v) 10.0

**452.** වෘත්තයක විශ්කම්භය / Diameter ඇතුළත් කළ විට එම වෘත්තයෙහි වර්ගපලය ලබා ගැනීමට සකස් කරන ලද කේතය පහත වේ. එහි ඇතුළත් දෝෂ වර්ගය වන්නේ,

```
A = int(input('Enter Diameter: '))
Area = (22/7)*(A/2)*(A/2)
Print(Area)
```

- i) Semestry Error                                      ii) Logic Error                                      iii) Syntax Error
- iv) All    v) No errors in the Code

**453. RPM (rotaion per minite)** යනු විනාඩි එකකදී යම් කිසි වස්තුවක් භ්‍රමණය වන වට ගණන වේ. වාහනයක රෝදයක පරිදිය / circumference = 1m වේ. එම වාහනයේ RPM එක ලබා දුන් විට එහි වේගය පැයට කිලෝමීටර ( $Kmh^{-1}$ ) කොපමණද යන්න ප්‍රතිදානය සඳහා කේතය පහත වේ. එහි A හිස්තැනට සුදුසුම කේත ජේලිය තෝරන්න.

```
C = 1 # Wheel circumference = 1m = 100cm = 0.001 km
RPM = float(input('Enter RPM of the Wheel :'))
Kmph = ..... ← A
print('Speed = ', Kmph, ' Kmph')
```

- i)  $1000 * RPM * 60$                                       ii)  $(RPM * C) / 1000 * 60$                                       iii)  $RPM * 60 / 1000$
- iv)  $RPM * 60 * 0.01$                                       v) i and iii both

**454.** ඉහත කේතය වෙනට 200 ආදානය කළ විට ලැබෙන ප්‍රතිදානය වන්නේ,  
 i) 120                      ii) 200                      iii) 120.0                      iv) '220'                      v) 12

**455.** පැය 12 හා පැය 24 ලෙස වේලාව ප්‍රකාශ කරන ආකාර 2ක් පවතී. (2310 → 1110pm) පැය 24 ක්‍රමයෙන් ආදානය කරන වේලාව 12 ක්‍රමයට හරවා එය ප්‍රතිදානය කිරීමට කේතය පහත වේ. එහි නිර්වද්‍යතාවය නවම පරීක්ෂාකර නොමැත. Syntax Errors නොපවතින බව නහවුරුයි.

```
T24 = int(input('Enter Time in 24 : '))
T12 = 0
Time = str()
if T24 >= 1200 and T24 <= 1300:
    Time = str(T24)+'pm'

elif T24 > 1200:
    T12 = T24 - 1200
    Time = str(T12)+'pm'

else :
    Time = str(T24)+'am'
print(Time)
```

ඉහත කේතය සඳහා 1258 යන පැය 24න් ලියන ලද වේලාව ඇතුළත් කළ විට ප්‍රතිදානය වන්නේ,  
 i) 1258am                      ii) 12.58pm                      iii) 1258 pm                      iv) 1258pm                      v) Error

**456.** ඉහත 449 කේතය සඳහාම 1300 ආදානය කළ විට ප්‍රතිදානය වන්නේ,  
 i) 1300am                      ii) 1300pm                      iii) 1pm                      iv) 1am                      v) Error

457. ඉහත 456 කේතයේ දෝෂ පවතී ද? පවතී නම් එය කුමන වර්ගයේ දෝෂයක් ද?

- i) No any type of Errors
- ii) Logical Error
- iii) Semantic Errors
- iv) Logical & Semantic Error
- v) Error

458. පහත කේතයෙහි කාර්යය / task වන්නේ,

```
Inp = int (input ('Enter a Number :'))
O = open('a.txt', 'a')
P = open('b.txt', 'a')
if Inp % 2 == 0 :
    O.write(str(Inp))
else :
    P.write(str(Inp))
O.close( ) ; P.close( )
```

- i) ආදානය අගය ඉරට්ටේ ඔත්තේ ලෙස වෙන් වෙන්ව ගොනුවක ගබඩා කිරීම.
- ii) ආදානය අගය ඉරට්ටේ ඔත්තේ ලෙස වෙන් වෙන්ව ගොනුවක 2 ගබඩා කිරීම.
- iii) ආදාන අගයන් මාරුවෙන් මාරුවට ගොනු දෙකට බෙදීම
- iv) ආදානයන්ගේ ප්‍රතිශත අගය ගොනුවක ගබඩා කිරීම
- v) ඉහත කිසිවක් නොවේ.

459. පහත ඇල්ගොරිතමය සලකා පිළිතුරු සපයන්න.

```
Q , T = 0 , 1
while Q < T :
    T = int (input ('Enter a Number :'))
    if Q > T :
        T = Q
    print ('Hi')
```

- i) 10ට වැඩි ප්‍රතිදාන සඳහා ලෙස ප්‍රතිදාය කරවන අතර ඊට අඩු ඒවා මගින් කේතය අවසන් කරයි.
- ii) - අගයන් වලට ලෙසද + අගයන්ට ද ලෙස ප්‍රතිදානය කරවයි.
- iii) කුමන ප්‍රතිදානය ආදානය කළද නොනැවතී ලුපය ගමන් කරයි.
- iv) - අගය ඇතුළත් කළ විට ලුපය නවතී.
- v) ඉහත කිසිවක් නොවේ.

>>> කැම කඩයක් සඳහා නිපදවූ මෘදුකාංගයක් පහත වේ. මෙම කැම කඩයේ කැම වර්ග 5ක් ඇත. එයට නියමිත අනන්‍ය අංකයක්ද ඇත. ඒ ඇහුරෙන් 460 සිට 463 දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න. පළමුව අදාළ පාරිභෝගිකයා මිලදී ගත් ආහාර ගණන ඇතුළත් කරයි. අනතුරුව එක් එක් ආහාරට අයත් අංකය ඇතුළත් කරයි. ඉන් පසු ගෙවිය යුතු මුදල මුදල ඇතුළත් කරයි.

කැම අංකය සහ එහි මිල : (0 – Rs.40 | 1 – Rs.35 | 2 – Rs.45 | 3 – Rs.50 | 4 – Rs.20)

```
Price = [40, 35, 45, 50, 20] ; Total = 0
K = int(input('How many Foods did you buy :'))
for H in range(.....) : ← A
    J = int(input('Enter Food Number : '))
    Total = Total + ..... ← B
print('Your Total Price : ', 'Rs.', Total)
```

460. ඉහත A හිස්තැනට වඩාත්ම සුදුසු පිළිතුර වන්නේ,  
 i) K                      ii) len(Price)                      iii) Price                      iv) J                      v) K-1

461. ඉහත B හිස්තැනට වඩාත්ම සුදුසු පිළිතුර වන්නේ,  
 i) Total                      ii) J                      iii) Price[ J ]                      iv) Price[ J + 1 ]                      v) K

462. කමලා කෑම වර්ග 6ක් මිලදී ගන්නාය. එවා නම් Rs.40 යේ රෝල්ස් 2ක්, Rs.45 එළවලු රොටි 3ක් හා Rs.20 වඩේ එකකි. මෙහි මුලු එකතුව ලබා ගැනීමට නිවැරදි ආදාන ලබා දිය යුතු ආකාරය පිළිවලින් වන්නේ,

- i) 2 2 2 4 4 6                      ii) 6 2 2 2 2 0 4                      iii) 6 2 0 2 4 2 0  
iv) 6 2 2 2 4 4 0 0                      v) 6 6 2 2 2 4 4 0

463. ආහාර සැලේ සෑම Rs.300 වැඩි බිල් පතක් සඳහාම 10% ක වට්ටමක් ලබා දේ. එම වට්ටමද සහිතව මුලු එකතුව [Total] ප්‍රතිදානයට [print] කේතයේ, print('Your Total Price : ', 'Rs.', Total) කේත ඡේද්‍රියට පෙර සිදු කළ යුතු වෙනස් කම වන්නේ,

- i) if Total > 300 :                      ii) if Total > 300 :  
    Total = Total + Total \* 0.1                      Total = Total - Total \* 0.1  
iv) if Total > 300 :                      iii) if Total > 300 :  
    Total = Total - Total / 100 / 10                      Total = Total - Total \* 100 / 10  
v) All Correct

464. පහත කේතයේ ප්‍රතිදානය / output වන්නේ,

```
def Mes (m) :
    print(m*4, end = ' ')
    return m
X = Mes(6)
if X > 20 :
    print('Py')
```

- i) 24                      ii) 24 Py                      iii) Py24                      iv) 24Py                      v) Error

465. පහත කේතයේ ප්‍රතිදානය / output වන්නේ, input = 5

```
U = int (input ('Enter Value :')) ; Total = 0
for k in range (U):
    Total = Total * ( k )
print(Total)
```

- i) -1                      ii) 0                      iii) 120                      iv) 24                      v) Error

466. පහත කේතයේ ප්‍රතිදානය / output වන්නේ, ආදානය ලෙස ' G ' ඉංග්‍රීසි අක්ෂරය ඇතුළත් කරන ලදී.

```
P = input('Enter a any English Letter :')
Z = P
if Z == P.upper() :
    print('C')
else:
    print('S')
```

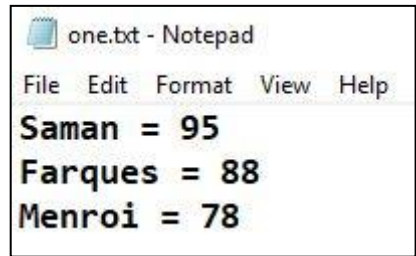
- i) S                      ii) CS                      iii) No outputs                      iv) C                      v) Error

467. ඉහත කේතයේ කාර්යය / task වන්නේ,

- i) ආදානය කරන අක්ෂරය P නම් C ලෙසද Z නම් S ලෙසද ප්‍රතිදානය කිරීම.  
ii) ආදානය කරන අක්ෂරය Capital නම් C ලෙසද Simple නම් S ලෙසද ප්‍රතිදානය කිරීම.  
iii) ආදානය කරන අක්ෂරය C නම් p ලෙසද Z නම් s ලෙසද ප්‍රතිදානය කිරීම.  
iv) ආදානය කරන අක්ෂර ප්‍රතිදානය කිරීම.  
v) විශේෂිත වූ කාර්යයක් නැත.

468. පහත කේතය සම්බන්ධයෙන් නිවැරදි ප්‍රකාශය වන්නේ, **one.txt** ගොනුව පහත දැක්වේ.

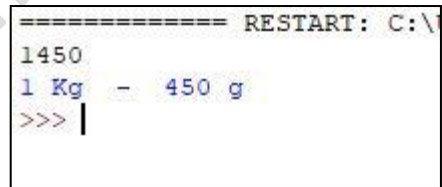
```
One = open ('one.txt' , 'r')
Two = open ('two.txt' , 'w')
Two.write(One.read ( ))
One.close( )
Two.close( )
```



- i) සම්පූර්ණ දෝෂ සහිත කේත කොටසකි.
- ii) ගොනු විවෘත කර වැසීම සිදු වන අතර අමතරව කිසිදු කාර්යයක් සිදු නොවේ.
- iii) **one.txt** ගොනුව කියවා එහි දත්ත **two.txt** ගොනුවෙහි ලිවීම.
- iv) ගොනුවට දත්ත වලින් ගත් කළ සමාන වෙනත් නම් ගොනුවක් නිර්මාණය කිරීම.
- v) ගොනු දෙකෙහි දත්ත සමාන දැයි පරීක්ෂා කිරීම.

469. තරාදියක් සඳහා එහි මෘදුකාංග කොටස නිපදවීමට පයිතන් යොදාගත්තා යැයි සිතන්න. තරාදියේ පවතින බර මැනීම සම්බන්ධ සංවේදකයේ **0 – 10kg** අතර කුඩාම අගය **1g** වන ලෙස මැන ගත හැකිය. සංවේදකය මගින් පරිගණකයට එහි මත තබා ඇති ස්කන්දයේ අගය සෑම විටම ලබා ඇතුළත් කරයි. එම අගය ලබා දෙනුයේ සංවේදකය මත **1kg** තැබූවිට **1000** ලෙසද **1.25** තැබූ විට **1250** ලෙසය. එනම් ග්‍රෑම් ගණන වේ. සංවේදකයේ ආදානය ගෙන කිලෝග්‍රෑම් හා ග්‍රෑම් ලෙස වෙන වෙනම බර ප්‍රතිදානය සඳහා කේතය පහත වේ. ප්‍රතිදානය රූපයේ පරිදි.

```
SensorIn = int (input ())
Kg = SensorIn // 1000
..... ← A
print (Kg , 'Kg' , ' - ' , g , 'g')
```



A හිස්තැනටා ගැලපෙන කේත කොටස වන්නේ.

- i)  $SensorIn = Sensor - g$
- ii)  $Kg = Sensor - g$
- iii)  $g = 1000 - kg / 1000$
- iv)  $g = SensorIn \% 1000$
- v)  $g = Kg - g$

470. ඉහත 469 ප්‍රශ්නය හී කේතයම වැඩි දියුණු කිරීමක් සිදු කළහ. එනම් **1kg**ක මිල ඇතුළත් කළවිට බර සමග බරට අදාළ මිලද ප්‍රතිදානය වීමට සැලැස් වීමයි. එහි කේතය පහත වේ. එහි **B** ගැලපෙන පිළිතුර වන්නේ,

```
SensorIn = int (input ())
K = int (input ('Enter Price of a Kilogram (Kg) :'))
Kg = SensorIn // 1000
..... ← A
print (Kg , 'Kg' , ' - ' , g , 'g')
..... ← B
print('Rs.' , Price)
```

- i)  $Price = SensorIn / 1000 * K$
- ii)  $Price = Kg + g / 1000 * K$
- iii)  $Price = Kg / 1000 + g / 1000 * K$
- iv) i and ii
- v) i and iii

471. ඉහත කේතය සඳහා ව්‍යාජ කේතය ලියන්න.

>>> යම් කිසි මුදල් ප්‍රමාණයක් ඇතුළත් කළ විට එය ගෙවීමට භාවිතා කළ යුතු නෝට්ටු සහ කාසි වල අගයන් ප්‍රතිදානය / **output** අදාළ කේතය පහත වේ. ඒ ඇසුරෙන් **472** සිට **474** දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
Cost = int (input ('Enter a Cost '))
C = (5000, 2000, 1000, 500, 100, 50, 20, 10, 5, 2, 1)
A , Count = True , 0
while Cost > 1 :
    if Cost >= C[Count] :
        print(C[Count], end = '|')
        Cost = Cost - C[Count]
    else:
        Count = Count + 1
```

**472.** ඉහත C නම් list එකේ සඳහන් වන්නේ මොනවාද,

- i) ශ්‍රීලංකාවේ මුදල් නෝට්ටු හා කාසි                      ii) මුදල් ගෙවීමට හැකි ප්‍රමාණ                      iii) අහඹු අගයන්  
iv) මුදලක් බෙදෙන ආකාරය                      v) නිශ්චිතව කිව නොහැකිය.

**473.** ආදානය ලෙස 4555 යන මුදලා ආදානය කරන ලදී. එවිට ප්‍රතිදානය වන්නේ,

- i) 2000|1000|1000|1000|500|50|5|                      ii) 2000|2000|500|50|2|2|1  
iii) 4000|500|50|5                      iv) 2000|2000|500|50|5|  
v) Error

**474.** ආදානය ලෙස 502 යන මුදලා ආදානය කරන ලදී. එවිට ප්‍රතිදානය වන්නේ,

- i) 500|2|                      ii) 200|200|100|2                      iii) 100|100|100|100|100|2  
iv) 2000|2000|500|50|5|                      v) 500|2

**475.** ඉහත පරිදි සැකසුණු සරල මෘදුකාංගයක භාවිතයක් වන්නේ,

- i) වෙලදසැලක භාණ්ඩ වල මිල ලබා ගෙන ඉතිරි මුදල පහසුවෙන් ලබා දීම.  
ii) බැංකුවක පාරිභෝගිකයා ඉල්ලුම් කරන මුදල පහසුවෙන් ඔහුට ලබා දීම.  
iii) ව්‍යාපාරයේ මුදල් ගිණුමක කිරීමට.  
iv) **i and ii**  
v) **ALL**

**476.** පහත කේතය සඳහා 1100101 ආදානය කළ විට ප්‍රතිදානය වන්නේ,

```
Binary = input ('Enter a Binary Value: ')
Decimal = int (Binary, 2)
print(Decimal)
```

- i) 102                      ii) 101.0                      iii) 101                      iv) 213                      v) Error

**477.** පහත කේතය සඳහා 457 ආදානය කළ විට ප්‍රතිදානය වන්නේ,

```
A = input ('Enter a 8 dec Value: ')
Decimal = int (A, 8)
print(Decimal)
```

- i) 4AD                      ii) 403                      iii) 313                      iv) 303                      v) Error

**478.** පහත කේතය සඳහා 6ABF ආදානය කළ විට ප්‍රතිදානය වන්නේ,

```
HD = input ('Enter a Hexadecimal Value: ')
Decimal = int (HD, 16)
print(Decimal)
```

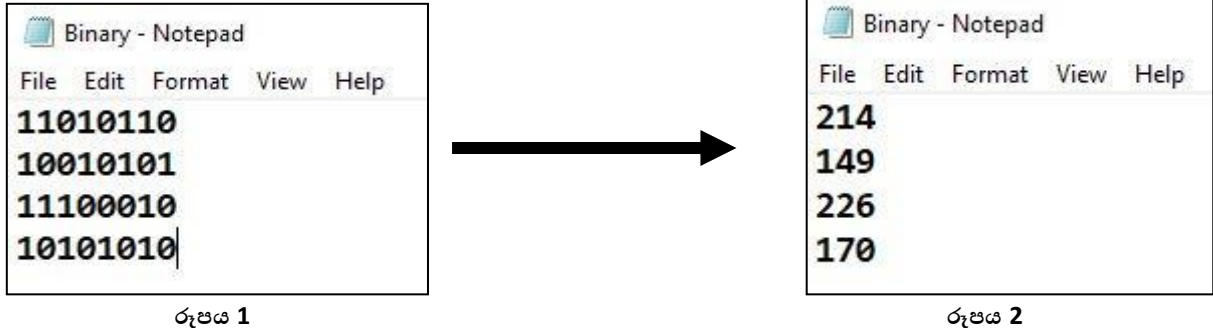
- i) 2727                      ii) 9909                      iii) Error                      iv) 27827                      v) 27327

479. පහත කේතය සඳහා 699 ආදානය කළ විට ප්‍රතිදානය වන්නේ,

```
P = input ('Enter a Value: ')
print(int(P, 0))
```

- i) 0000
- ii) 699
- iii) 0
- iv) 700
- v) Error

>>>. Binary.txt නම් ගොනුව පහත වේ. එහි 8bit Binary Values ගබඩා කර ඇත.



Binary.txt නම් ගොනුව රූපයේ 1 ආකාරයේ සිට රූපයේ 2 ආකාරයට වෙනස් කළ යුතුයි. වෙනස් කිරීමට, එනම් පළමුව තිබෙන අගයන්ට සමාන අගය ලෙස කිරීමට අවශ්‍යවී ඇත. එම කාර්යය සිදු කිරීමට සිසුවෙකු සැකසූ කේතයක් පහත වේ. ඒ ඇසුරෙන් 480 සිට 483 දක්වා ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
Decimal = []
Count = 0
.....Y...
Read = Bin.readline().strip()
while Read:
    Dec = int(Read, 2)
    Decimal.append(Dec)
    Read = Bin.readline().strip()
Bin.close()

.....Z...
for i in Decimal:
    Bin.write(str(i)+'\n')
Bin.close()
```

ඉහත කේතයේ A හා B ලෙස කේත කොටස් දෙකකින් පවතී,

480. A කොටසේ කාර්යය වන්නේ,

- i) Binary.txt කියවීම හා එහි කියවීම Decimal නම් වූ List එකක ගබඩා කර ගැනීම.
- ii) Binary.txt කියවා දත්ත Binary → Decimal හරවීම.
- iii) Binary.txt කියවා දත්ත එහි තුලම තාවකාලිකව ගබඩා කර තබා ගැනීම.
- iv) Binary.txt කියවා දත්ත පිළිවෙලකට සැකසීම.
- v) Binary.txt කියවා දත්ත Binary → Decimal ට හරවා ඒව Decimal නම් වූ List එකක ගබඩා කර ගැනීම.

481. B කොටසේ කාර්යය වන්නේ,

- i) A කියවා ගත් දත්ත Binary → Decimal හරවීම හා ඒවා Binary.txt තුළ ලිවීම
- ii) A හිදී සකසා ගත් දත්ත Binary.txt තුළින් ගෙන නැවත එහිම ලිවීම.
- iii) A කියවා ගත් දත්ත සැකසීමට ලක් කිරීම.
- iv) A හිදී Binary → Decimal හරවීම කළ දත්ත List එක තුළින් ගෙන නැවත Binary.txt ලිවීම.
- v) ඉහත කිසිවක් නොවේ.

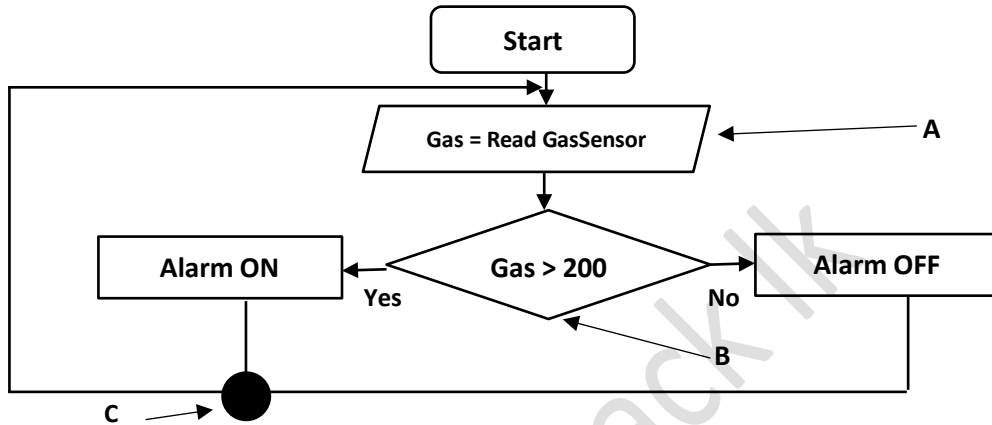
482. Y හිස්තැනට ගැලපෙන කේත ජේළිය විය හැක්කේ,

- i) Bin = open ('Binary.txt', 'w')
- ii) Read = write ('Bim.txt', 'r')
- iii) Bin = write ('Bim.txt', 'r')
- iv) Bin = open ('Binary.txt', 'r')
- v) Read = open ('Binary.txt', 'r')

483. Z හිස්තැනට ගැලපෙන කේත ජේළිය විය හැක්කේ,

- i) Bin = open ('Binary.txt', 'w')
- ii) Bin = open ('Bim.txt', 'a')
- iii) Decimal = write ('Bim.txt', 'r+')
- iv) Read = open ('Binary.txt', 'w')
- v) dec = open ('Binary.txt', 'w+')

>>> නිවසක ගැස් කාන්දුවක් හඳුනාගැනීම සඳහා ගැස් සංවේදකයක් මගින් කියවීම්/ Sensor Analog Readings ලබා ගනී. ගැස් කාන්දුවකදී එහි අගය 200 ට වඩා ඉහළ යන අතර සාමාන්‍ය අවස්ථා වලදී 200 අඩුවෙන් පවතී. පැය 24 පුරාවටම ක්‍රියාත්මක වේ. එම පද්ධතිය සඳහා ඇල්ගොරිතමය පහත වේ. ඒ ඇසුරෙන් 484 හා 485 ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.



484. මෙම ඇල්ගොරිතමයේ අවසානය / stop යෙදිය යුත්තේ කොතැනද,

- i) A
- ii) B
- iii) C
- iv) A and C
- v) No Need Stop

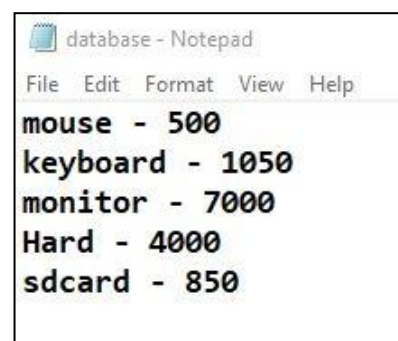
485. C හි පවතින්නේ කුමක්ද,

- i) Decision
- ii) Plusing
- iii) Connector
- iv) Procces
- v) Comment Box

>>> පහත දක්වා database.txt යන්නහි අන්තර්ගතයි. පළමුව භාණ්ඩ දේවනුව මිලද පවතී. භාණ්ඩයේ නම ඇතුළත් කළ විට මිල දර්ශනය කිරීම කේතය මගින් සිදු කරයි. ඒ ඇසුරෙන් 487 - 490 ප්‍රශ්න වලට පිළිතුරු සපයන්න.

```

data = open('database.txt', 'r')
G = input("Enter Good's Name :")
Read = data.readline().strip()
while Read:
    D = .....X....
    if D[0] == G:
        print(D[1])
    Read = data.readline().strip()
data.close()
    
```



486. ඉහත A මගින් පෙන්වන .strip() යන්නහි අවශ්‍යතාවය වන්නේ,

- i) database.txt සිට කියවන දත්තය නම සහ මිල ලෙස වෙන් කර ගැනීම.
- ii) database.txt සිට කියවන දත්තයහි දෙපස ඇති අනවශ්‍ය දේ ඉවත් කිරීම.
- iii) දත්තයේ නිරවද්‍යතාවය තහවුරු කිරීම.
- iv) කියවා ගත් දත්ත ජේළිය String Type එකට හැරවීම සඳහා.
- v) ඉහත කිසිවක් නොවේ.







500. පහත කේතයේ ප්‍රතිදානය වන්නේ,

```
e = 0
for i in range(5):
    for k in range(6):
        for j in range(7):
            e = e + 1
print(e)
```

- i) 540                      ii) 240                      iii) 177                      iv) 210                      v) Error

501. පහත කේතයේ ප්‍රතිදානය වන්නේ,

```
h=1
for i in range(h):
    h = i
    print('b')
```

- i) 'b' එකවරක්                      ii) 'b' දෙවරක්                      iii) 'b' හතර වරක්                      iv) 'b' නොනවත්වා                      v) Error

502. පයිතන් භාෂාව ලොවට එළිදැක්වූ වර්ෂ,

- i) 1990                      ii) 1996                      iii) 1989                      iv) 1978                      v) 1999

>>> පහත කේතය සලකා 497, 498 ප්‍රශ්න සඳහා පිළිතුරු සපයන්න.

```
K = int (input ('Enter No :'))
K = K / 0.5
K = K * 8
K = K + 9
print (K)
```

503. ඉහත කේතයට 34 ආදානය කළ විට ප්‍රතිදානය වන්නේ,

- i) 778.0                      ii) 35.0                      iii) 500.00                      iv) 533.0                      v) 199.9

504. ඉහත කේතයට 0 ආදානය කළ විට ප්‍රතිදානය වන්නේ,

- i) 0                      ii) 0.0                      iii) 9.0                      iv) 9                      v) Error

505. පහත ව්‍යාජ කේතය සලකන්න. එහි ප්‍රතිදානය වන්නේ,

```
Begin
    total = 0
    for count = 1 To 10
        if count is odd Then
            total = total + count
        end if
    Next count
    Display total
End
```

- i) 10                      ii) 15                      iii) 20                      iv) 25                      v) 55

YouTube.com / ideaPack lk

Learn Arduino | Electronics | ICT | Science | Technology

# Answers

1. 3	33.4	66.1
2. 3	34.2	67.2
3. 4	35.3	68.3
4. 4	36.4	69.1
5. 1	37.2	70.3
6. 4	38.3	71.1
7. 5	39.3	72.1
8. 1	40.4	73.4
9. 2, 5	41.5	74.1
10.1	42.4	75.1
11.1	43.5	76.1
12.3	44.4	77.4
13.4	45.1	78.4
14.3	46.4	79.4
15.3	47.5	80.2
16.1	48.4	81.2
17.3	49.2	82.4
18.1, 3	50.1	83.1
19.2	51.3	84.2
20.5	52.5	85.5
21.3	53.1	86.3
22.4	54.2	87.3
23.2	55.4	88.5
24.2	56.2	89.2
25.1	57.1	90.3
26.3	58.1	91.4
27.3, 5	59.2	92.1
28.5	60.4	93.3
29.5	61.1	94.2
30.5	62.2	95.4
31.4	63.4	96.1
32.2	64.3	97.2, 3
	65.4	98.2

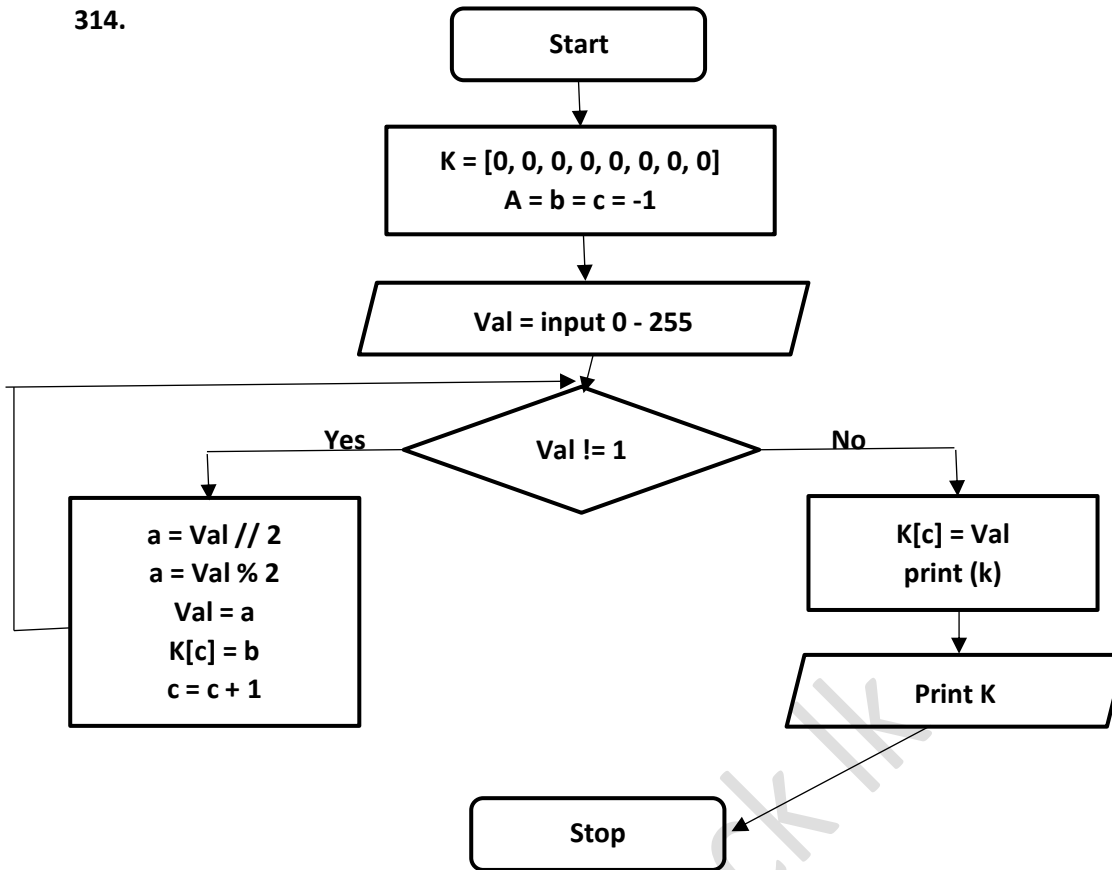
99.4		134.	1	169.	5
100.	1, 4	135.	5	170.	1
101.	3	136.	5	171.	2
102.	1	137.	2	172.	3
103.	5	138.	3	173.	2
104.	4	139.	4	174.	3
105.	4	140.	5	175.	2
106.	2	141.	1	176.	5
107.	2	142.	1	177.	5
108.	2	143.	4	178.	2
109.	1	144.	3	179.	5
110.	1	145.	1	180.	1
111.	4	146.	1	181.	4
112.	4	147.	1	182.	3
113.	3	148.	5	183.	1
114.	2	149.	5	184.	1
115.	4	150.	4	185.	2
116.	2	151.	1	186.	3
117.	5	152.	4	187.	1
118.	4	153.	2	188.	1
119.	5	154.	4	189.	5
120.	3	155.	4	190.	1
121.	1	156.	3	191.	4
122.	3	157.	2	192.	5
123.	3	158.	4	193.	4
124.	1	159.	2	194.	5
125.	1	160.	1	195.	1
126.	2	161.	5	196.	3
127.	3	162.	1	197.	2
128.	1	163.	1	198.	3
129.	4	164.	2	199.	1
130.	3	165.	3	200.	4
131.	4	166.	4	201.	5
132.	3	167.	5	202.	2
133.	1	168.	5	203.	2

204.	1	239.	5	274.	3
205.	2	240.	3	275.	3
206.	3	241.	2	276.	2
207.	5	242.	2	277.	1
208.	3	243.	5	278.	4
209.	5	244.	1	279.	3
210.	5	245.	3	280.	1
211.	1	246.	4	281.	3,4
212.	3	247.	1	282.	3
213.	5	248.	5	283.	3
214.	3	249.	5	284.	1
215.	1	250.	5	285.	4
216.	1	251.	3	286.	1
217.	2	252.	4	287.	4
218.	4	253.	2	288.	2
219.	3	254.	3	289.	1
220.	4	255.	5	290.	2
221.	3	256.	5	291.	2,3
222.	5	257.	3	292.	5
223.	2	258.	4	293.	4
224.	5	259.	2	294.	3
225.	3	260.	1	295.	1
226.	4	261.	4	296.	2
227.	5	262.	2	297.	2
228.	5	263.	2	298.	3
229.	5	264.	4	299.	1
230.	5	265.	2	300.	2
231.	1	266.	3	301.	5
232.	1	267.	3	302.	3
233.	5	268.	4	303.	4
234.	4	269.	1	304.	1
235.	1	270.	1	305.	3
236.	2	271.	3	306.	3
237.	3	272.	2	307.	5
238.	1	273.	3	308.	4

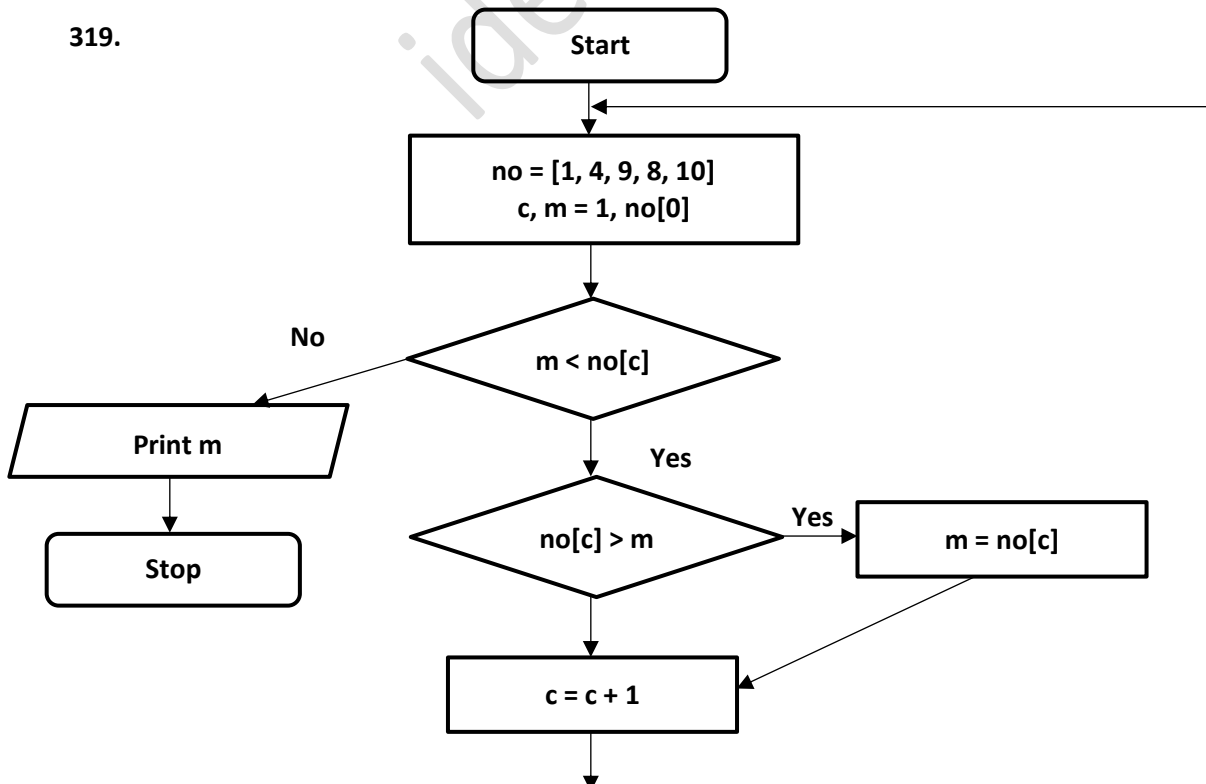
309.	4	344.	3	379.	1
310.	5	345.	1	380.	5
311.	1	346.	Last Page	381.	2
312.	3	347.	2	382.	4
313.	5	348.	1	383.	2
314.	Last Page	349.	5	384.	3
315.	5	350.	4	385.	Last Page
316.	5	351.	3	386.	5
317.	2	352.	1	387.	2
318.	3	353.	2	388.	4
319.	Last Page	354.	5	389.	3
320.	4	355.	2	390.	2
321.	3	356.	5	391.	3
322.	4	357.	Last Page	392.	2
323.	1	358.	5	393.	5
324.	5	359.	1	394.	4
325.	2	360.	5	395.	2
326.	4	361.	3	396.	3
327.	5	362.	3	397.	3
328.	3	363.	2	398.	5
329.	2	364.	4	399.	3
330.	4	365.	4	400.	5
331.	3	366.	2	401.	4
332.	1	367.	4	402.	4
333.	2	368.	4	403.	2
334.	1	369.	3	404.	2
335.	5	370.	Last Page	405.	4
336.	5	371.	3	406.	2, 4
337.	3	372.	1	407.	5
338.	4	373.	3	408.	2
339.	Last Page	374.	1	409.	4
340.	2	375.	2	410.	3
341.	1	376.	2	411.	1
342.	2, 4	377.	5	412.	2
343.	4	378.	3	413.	5

414.	4	449.	2	484.	5
415.	4	450.	2	485.	3
416.	Last Page	451.	5	486.	2
417.	4	452.	3	487.	3
418.	3	453.	2	488.	4
419.	1	454.	5	489.	4
420.	4	455.	4	490.	5
421.	4	456.	2	491.	3
422.	3	457.	2	492.	1
423.	4	458.	2	493.	5
424.	2	459.	4	494.	5
425.	1	460.	1	495.	2
426.	3	461.	3	496.	2
427.	3	462.	3	497.	2
428.	1	463.	2	498.	4
429.	4	464.	1	499.	2
430.	2	465.	2	500.	4
431.	2	466.	4	501.	1
432.	2	467.	2	502.	4
433.	3	468.	3, 4	503.	4
434.	3	469.	4	504.	3
435.	4	470.	1	505.	4
436.	2	471.	Last Page		
437.	3	472.	1		
438.	2	473.	4		
439.	1	474.	1		
440.	3	475.	4		
441.	2	476.	3		
442.	1	477.	4		
443.	1	478.	5		
444.	2	479.	2		
445.	Last Page	480.	1		
446.	4	481.	1		
447.	4	482.	4		
448.	1	483.	2		

314.

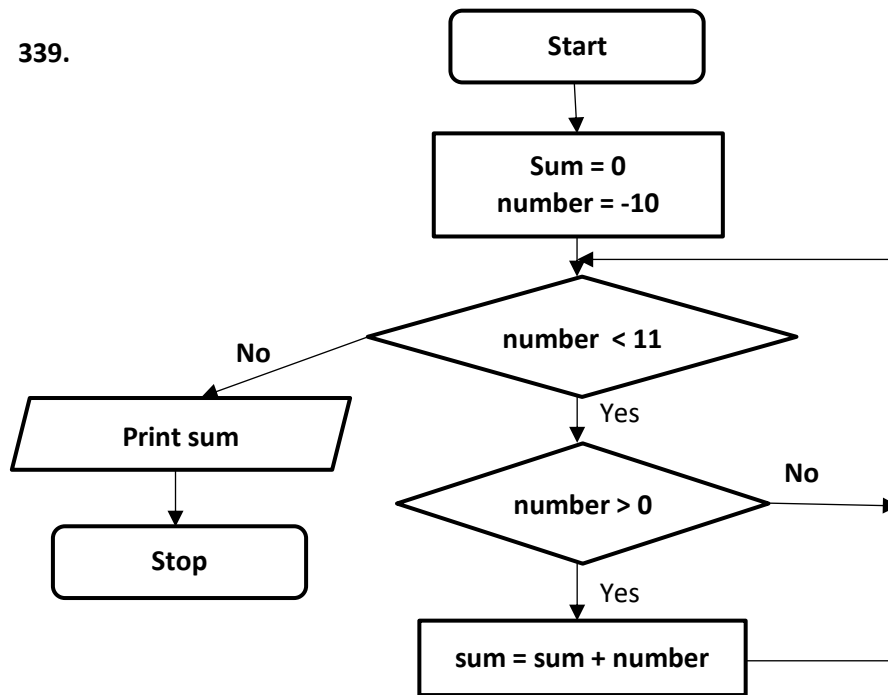


319.





339.



346.

BEGIN

```

A = input integer value
B = input integer value
C = A * B
If C >= 1 then C = 1
Else C = 0
END IF
Display C
  
```

END

357.

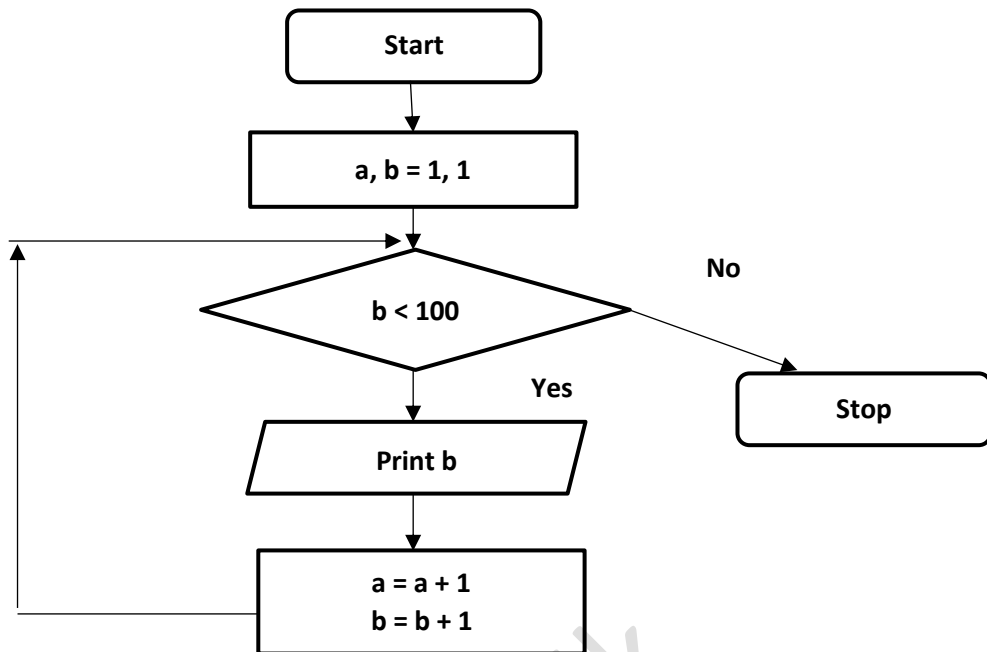
Begin

```

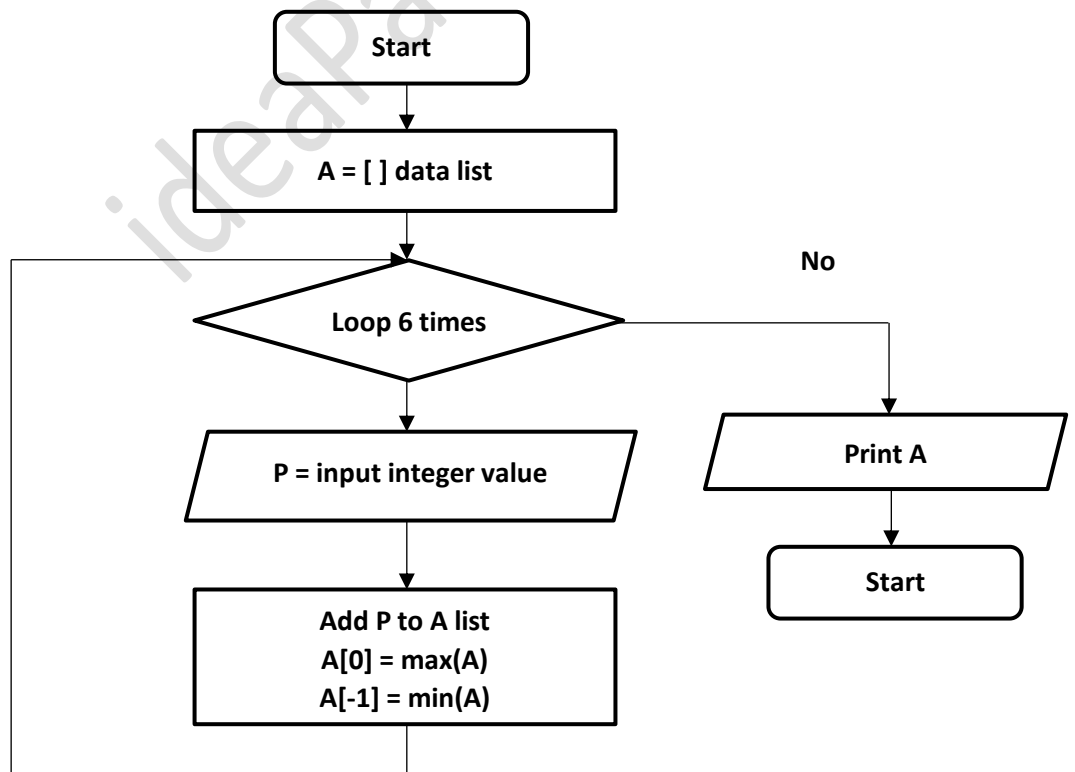
A = input Vale
B = A.split(',')
C = A[0]
If C >= 240 then print 'E'
Else If C >= 224 then print 'D'
Else If C >= 192 then print 'C'
Else If C >= 128 then print 'B'
Else If C >= 0 then print 'A'
  
```

Stop

370.



416.



385.

BEGIN

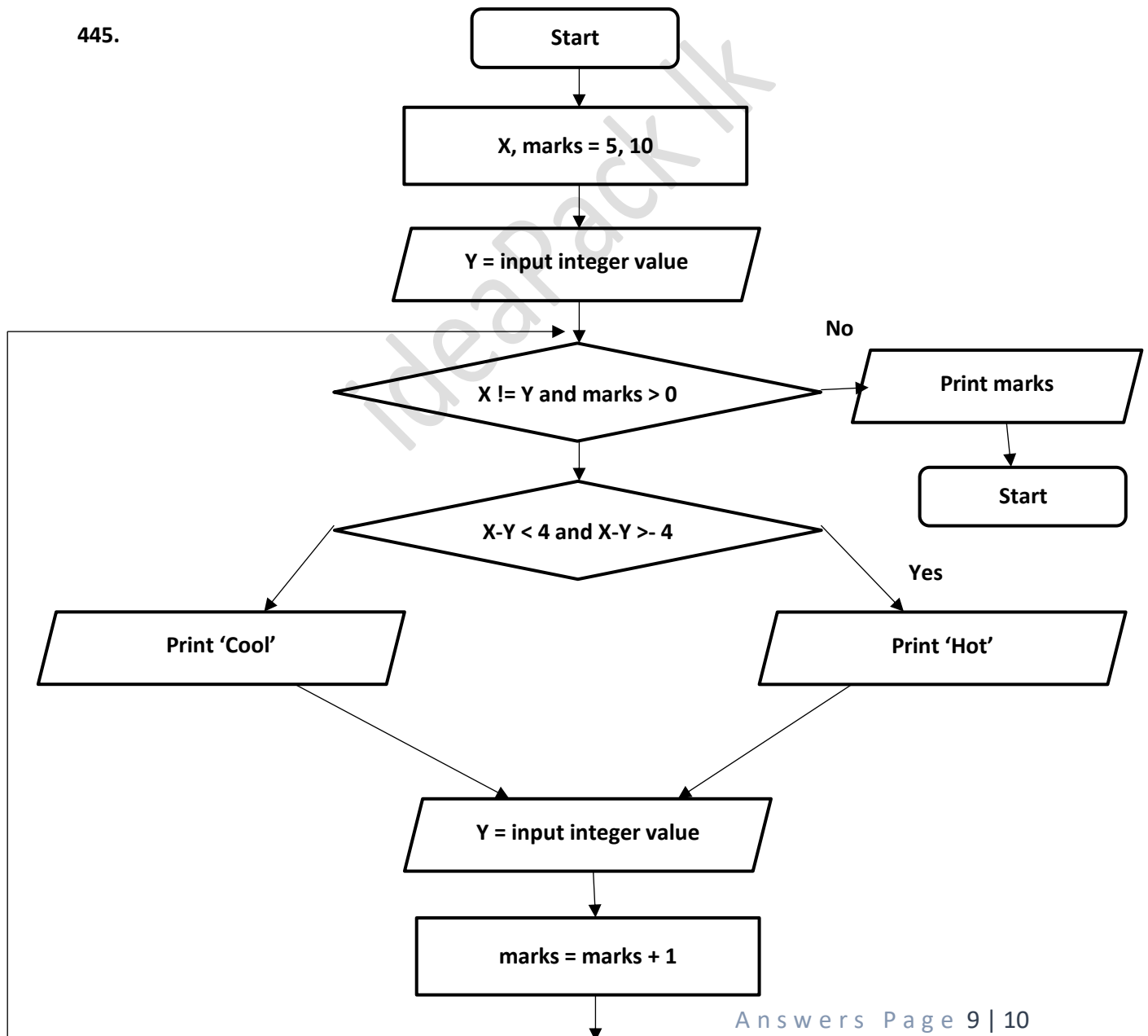
```

count = 0
for i looping len(A) times do
    B = A[i]
    If B % 2 == 0 then do
        count = count + 1
    else
        count = count + 1
    END IF
Disply count
END FOR LOOP

```

END

445.



471.

**BEGIN**`SensorIn = 0``K = input integer value``Kg = SensorIn // 1000``g = SensorIn % 1000``print(Kg, g)``Price = SensorIn / 1000 / * K`**END**

Subscribe to our YouTube Channel : YouTube Search : ideaPack Ik

