

06 `x = (2**10/2**4.0 * (7/8))` ක්‍රියාත්මක කළ විට ප්‍රතිදානය වන අගය

- (1) 56.0
- (2) 0.0
- (3) 1.0
- (4) **SyntaxError**
- (5) තාර්කික දෝෂයක් පවතී.

07 පයිතන් උපදෙස් කාණ්ඩය ක්‍රියාත්මක කිරීමේ දී ප්‍රතිදානය වන දෝෂ ආකාරය සම්බන්ධයෙන් නිවැරදි ප්‍රකාශය කුමක් ද?

```
>>> a = 40
>>> c = 6
>>> d = a / (c - 6)
```

Traceback (most recent call last):
 File "<pysshell#11>", line 1, in <module>
 d = a / (c - 6)
ZeroDivisionError: integer division or modulo by zero

- (1) උපදෙස් ක්‍රියාත්මක කිරීමේ දී තාර්කික දෝෂ ආකාරයක් අන්තර්ගත වේ.
- (2) C සඳහා ආදේශිත අගය අනුව දෝෂ ආකාරය නිර්මාණය වී ඇත.
- (3) කාරක කාණ්ඩ අනුපිළිවෙල අනුව වාග්චික දෝෂ ආකාරය නිර්මාණය වී ඇත.
- (4) උපදෙස් ක්‍රියාත්මක කිරීමේ දී දෝෂය නිර්මාණය වී ඇත.
- (5) ඉහත සියල්ලම සත්‍ය වේ.

08 පයිතන් ක්‍රමලේඛ භාෂාවේ විචල්‍යය (Variable) භාවිතය අනුව නිවැරදි ප්‍රකාශය කුමක් ද?

- A. විචල්‍යය භාවිතයට පෙර, අගය ආදේශය සමඟ අර්ථ දැක්වීම අනිවාර්ය වේ.
 - B. විචල්‍යයයේ පවතින අගය අනුව දත්ත වර්ගය (Data Type) තීරණය වේ.
 - C. විචල්‍යයක් අදාළ වන දත්ත වර්ගය වෙනස් කළ නොහැක.
- (1) A පමණක් සත්‍ය වේ.
 - (2) B,C පමණක් සත්‍ය වේ.
 - (3) B පමණක් සත්‍ය වේ.
 - (4) A, B පමණක් සත්‍ය වේ.
 - (5) A,B,C සත්‍ය වේ.

09 නිවැරදි නොවන පයිතන් ප්‍රකාශය කුමක් ද?

- (1) `range(0, .5)`
- (2) `range(1, 3, 1)`
- (3) `range(10, 4, -1)`
- (4) `range(2, 4, -1)`
- (5) `range(1, 5, 5)`

10 පයිතන් උපදෙස් ක්‍රියාත්මක කිරීමේ දී ලැබෙන ප්‍රතිදානය කුමක් ද?

```
a = b = c = 20
c , b, a = 12, a, 16

print("{0} {2} {1}".format(a,b,c))
```

- (1) 12 20 16
- (2) 16 12 20
- (3) 20 16 12
- (4) **SyntaxError**
- (5) ඉහත කිසිවක් සත්‍ය නොවේ.

11 0.025 හි අගය නිවැරදිව දැක්වෙන්නේ

- (1) 2.5e2
- (2) 2.5e-2
- (3) 25e-2
- (4) 25e2
- (5) 0.025e2

12 ක්‍රමලේඛය අනුව `nPrint('K', 4)` ක්‍රියාත්මක කළ විට ලැබිය හැකි ප්‍රතිදානය කුමක් ද?

```
def nPrint(message, m):
    while m > 0:
        print(message),
        m -= 1
```

- (1) K K K K
- (2) KKKK
- (3) 4 4 4 4
- (4) 4444
- (5) ඉහත කිසිවක් සත්‍ය නොවේ.

- 13 `2 + 2 ** 3 / 2 // 2.0 % 2`
- (1) 3.0 (2) 3.0
 (3) 3.5 (4) 2.0
 (5) 0.0
- 14 `i, j = 10, i/2` පයිතන් උපදෙස් කාණ්ඩ ක්‍රියාත්මක කළ විට ලැබිය හැකි ප්‍රතිදානය
`i += i + j * 5` කුමක් ද?
`print("value :" + str(i))`
- (1) `value :str(45)` (2) `value :35`
 (3) `value :45` (4) දෝෂ ආකාරයක් ප්‍රතිදානය වේ.
 (5) ඉහත කිසිවක් නොවේ.
- 15 Python Competition ප්‍රකාශය දිස්වන වාරගණන වන්නේ කුමක් ද?
- ```
count = 0
while count < 10:
 print("Python Competition")
```
- (1) 9 (2) 10  
 (3) 0 (4) `SyntaxError`  
 (5) ඉහත කිසිවක් නොවේ.
- 16 පයිතන් ක්‍රමලේඛය ක්‍රියාත්මක කළ විට ප්‍රතිදානය ලබන වාර ගණන
- ```
for i in range(10):
    for j in range(10):
        print(i * j)
```
- (1) 10 (2) 9
 (3) 100 (4) 81
 (5) 99
- 17 ක්‍රියා පරිපාටියක (Function) ප්‍රත්‍යාගමන අගයක් (return value) දක්වා නොමැති විට ආවේනික අගය (default Value) වන්නේ කුමක් ද?
- (1) `null` (2) `None`
 (3) `int` (4) `False`
 (5) ඉහත කිසිවක් නොවේ.
- 18 ක්‍රියා පරිපාටි (Function) තුළ භාවිතා වන පරාමිතීන් (Parameters) සම්බන්ධයෙන් අසත්‍ය ප්‍රකාශය කුමක් ද?
- (1) සැකසීමට අවශ්‍ය වන අගයයන් පරිමිතක/තාර්කික අගයයන් ආකාරයෙන් ආදානය කරයි.
 (2) තාර්කික අගයයන් (logical Value) විශ්ව පරාමිතීන් (Global) ලෙස පවතී.
 (3) පරාමිතීක අගයයන් ගණන නියතයක් විය හැක.
 (4) විශ්ව පරාමිතීක අගයයක් ක්‍රියාපරිපාටියක් තුළ දී වෙනස් කළ හැකිය.
 (5) ක්‍රියාපරිපාටියක් නැවත නැවත ක්‍රියාත්මක කළ හැකි වේ.
- 19 `'p' in "Python"` අනුව ප්‍රතිදානය කුමක් ද?
- (1) `False` (2) `True`
 (3) `true` (4) `false`
 (5) ඉහත කිසිවක් නොවේ.
- 20 `num = 'String'` ක්‍රමලේඛය ක්‍රියාත්මක කළ විට ලැබෙන ප්‍රතිදානය
`num += num[0] * len(num[0])` කුමක් ද?
`print(num)`
- (1) `StringSSSSSS` (2) `S`
 (3) `StringS` (4) `String`
 (5) දෝෂ ආකාරයක් ප්‍රතිදානය කරයි.

- 21 අගය ලෙස 5 ප්‍රතිදානය වන ප්‍රකාශය කුමක් වන්නේ ද?
- (1) `int(4.9)` (2) `round(4.9)`
 (3) `int(5.5)` (4) `round(5.4)`
 (5) `int(5.5)` හා `round(5.4)`
- 22 **Tuple** දත්ත ව්‍යුහය සම්බන්ධයෙන් වඩාත්ම නිවැරදි ප්‍රකාශය වන්නේ
- (1) සංඛ්‍යාත්මක අගයයන් සමූහයක් තබාගැනීම සඳහා සුදුසු වේ.
 (2) අක්ෂර අඩංගු අගයයන් සමූහයක් අනුපිළිවෙල යටතේ තබාගත හැකි වේ.
 (3) අගයයන් වෙනස් කළ නොහැකි පරිදි තබාගැනීමට සඳහා වඩාත් සුදුසු වේ.
 (4) සංඛ්‍යාත්මක හා අක්ෂර ඇතුළත් අගයයන් මිශ්‍රණයක් තබාගත නොහැකි වේ.
 (5) ඉහත කිසිවක් සත්‍ය නොවේ.
- 23 **range(5)** මගින් ප්‍රතිදානය වන අගයයන් වන්නේ කුමක් ද?
- (1) `[0, 1, 2, 3, 4, 5]` (2) `0 1 2 3 4`
 (3) `[0, 1, 2, 3, 4]` (4) `0, 1, 2, 3, 4`
 (5) `0, 1, 2, 3, 4, 5`
- 24 ලයිතුවක් ආකාරයේ (**list**) දත්ත ව්‍යුහයක් නිර්මාණය කළ නොහැකි ප්‍රකාශය වන්නේ
- (1) `list1 = [0:6]` (2) `list1 = []`
 (3) `list1 = list([12, 4, 4])` (4) `list1 = [12, 4, 4]`
 (5) `list1 = [1, "CAD", "0142"]`
- 25 දත්ත ව්‍යුහයන් භාවිතය සම්බන්ධව වඩාත් නිවැරදි ප්‍රකාශය/ප්‍රකාශන වන්නේ කුමක් ද?
- A. List තුළ dictionary කිහිපයක් තබා ගත හැකිය.
 B. අගයයන් නොමැතිව ලයිස්තු (**List**) ආකාරයේ දත්ත පවතී.
 C. ලයිස්තුවක දත්ත වෙනස් කළ නොහැකි ආකාරයෙන් පවතී.
- (1) **A** පමණක් සත්‍ය වේ. (2) **B** පමණක් සත්‍ය වේ.
 (3) **A, B** පමණක් සත්‍ය වේ. (4) **B, C** පමණක් සත්‍ය වේ.
 (5) **A, B** හා **C** සත්‍ය වේ.
- 26 පුනරාවර්තන (**repetition**), **Loop** භාවිතය සම්බන්ධව වඩාත් නිවැරදි ප්‍රකාශය/ප්‍රකාශනයන් වන්නේ කුමක් ද?
- A. දන්නා වාර ගණනක් උපදෙස් කියාත්මක කිරීම සඳහා **while loop** භාවිතා කළ හැකිය.
 B. නොදන්නා වාර ගණනක් උපදෙස් ක්‍රියාත්මක කිරීම සඳහා **for loop** සුදුසු නොවේ.
 C. **loop** ප්‍රකාශණයක් අවශ්‍ය විටක පාලනය කළ හැකිය.
- (1) **A** පමණක් සත්‍ය වේ. (2) **B** පමණක් සත්‍ය වේ.
 (3) **A, B** පමණක් සත්‍ය වේ. (4) **B, C** පමණක් සත්‍ය වේ.
 (5) **A, B** හා **C** සත්‍ය වේ.
- 27 මිතුරන්ගේ නම් අනුව ඊමේල් ලිපින තබාගැනීම සඳහා භාවිතා කිරීමට සුදුසු දත්ත ව්‍යුහය කුමක් වේ ද?
- (1) **String** (2) **Tuple**
 (3) **Dictionary** (4) **List**
 (5) **List** සහ **Dictionary**
- 28 `d1 = [20,10]` නිර්මාණය කර තිබෙන දත්ත ලයිස්තු ආකාර අනුව වඩාත් ම නිවැරදි ප්‍රකාශය
`e1 = ['AB', 'CD']` කුමක් ද?
- (1) `d1 + e1` ක්‍රියාත්මක කළ විට `[20, 10, 'AB', 'CD']` ප්‍රතිදානය ලෙස සකස් වේ.
 (2) `d1 + e1` දත්ත ආකාර වෙනස් බැවින් දෝෂ ආකාරයක් ප්‍රතිදානය කරයි.
 (3) ප්‍රතිදාන අගය `['20AB', '10CD']` ලෙස සකස් වේ.
 (4) `d1 + d1` ක්‍රියාත්මක කළ විට `[40, 20]` ප්‍රතිදානය ලෙස සකස් වේ.
 (5) ඉහත කිසිවක් සත්‍ය නොවේ.

- 29 **Dictionary** දත්ත ව්‍යුහයක් නිර්මාණය කිරීම සම්බන්ධයෙන් දෝෂ පවතින ප්‍රකාශය කුමක් වේ ද?
- (1) `uoc = {d1:'Viraj', d2:'Roshan'}` (2) `empty = {}`
 (3) `uom = {10:"Nayana"}` (4) `uoc = {1:'Viraj', 2:'Roshan'}`
 (5) `uoc = {'1':'Sri', '2':'lanka'}`
- 30 `'22, 23, 24'.split(',')` ක්‍රියාත්මක කළ විට ප්‍රතිඵලය වන්නේ
- (1) `['22', '23', '24']` (2) `'22', '23', '24'`
 (3) `['22', '23', '24']` (4) `'22', '23', '24'`
 (5) `22 23 24`
- 31 **t** මගින් Tuple දත්ත ව්‍යුහයක් දැක්වෙන අතර, එම දත්ත ව්‍යුහයක් සමඟ ක්‍රියාත්මක නොවන ප්‍රකාශය කුමක් වේ ද?
- (1) `t = ('ICT', 21, 2017)` (2) `t = t.sort()`
 (3) `t[0:2]` (4) `print(len(t))`
 (5) `print(t[0])`
- 32 පයිතන් ක්‍රමලේඛ තුළ ගොනු (File) භාවිතය අනුව නිවැරදි ප්‍රකාශය කුමක් ද?
- (1) දත්ත කියවීම (read) සඳහා ගොනුව විවෘත කිරීමකදී, ගොනුව පිහිටුම තුළ නොපවතින විට දෝෂ නිර්මාණය වේ.
 (2) දත්ත ලිවීම (write) සඳහා ගොනුව විවෘත කිරීමකදී, ගොනුව පිහිටුම තුළ නොපවතින විට දෝෂ නිර්මාණය වේ.
 (3) දත්ත කියවීම සඳහා ගොනුව විවෘත කිරීමකදී, ගොනුව පිහිටුම තුළ නොපවතින විට, වැඩසටහන මගින් ගොනුවක් නිර්මාණය කරයි.
 (4) දත්ත ලිවීම සඳහා ගොනුවක් විවෘත කිරීමකදී, ලියන දත්ත ගොනුවේ ඕනෑම පිහිටුමක අඩංගු කළ නොහැකිය.
 (5) දත්ත ලිවීමකදී එම දත්ත ගොනුව ආරම්භයේ සිට පමණක් අන්තර්ගත වේ.
- 33 ගොනුවක් කියවීම සඳහා භාවිතා වන පහත දැක්වෙන කුමන පයිතන් (Python) ක්‍රමලේඛ කණ්ඩාය කාරක නීති අනුව නිවැරදි (syntactically correct) වේද?
- (1) `file = open('marks.txt','a')`
`data = file.readlines()`
`for line in data:`
`print(line)`
 (2) `file = open('marks.py','r')`
`data = file.readlines()`
`for line in data:`
`print(line)`
 (3) `file = open('marks.txt','r')`
`data = file.readlines()`
`for line in data: print(line)`
 (4) `file = open('marks.txt','r')`
`data = file.readlines()`
`for line in data;`
`print(line)`
 (5) `file = open('marks.txt','r')`
`data = file.readlines():`
`for line in data:`
`print(line);`
- 34 පහත දැක්වෙන පයිතන් Python දත්ත ප්‍රථම අයිතම සලකා බලන්න.
- A. `'22.2'`
 B. `[2015, 'ict', 05]`
 C. `{ 'name' : 'Nimal', 'age' : 18}`
 D. `([22,33], [0,2])`
- ඉහත A, B, C දත්ත අයිතම වල Python දත්ත ප්‍රථම (Data types) පිළිවෙලින්
- (1) `String, list, dictionary, List` (2) `number, list, dictionary, List`
 (3) `String, list, dictionary, Tuple` (4) `number, list, dictionary, Tuple`
 (5) ඉහත කිසිවක් නොවේ.
- 35 `10-4*10/2**5 * 2` යන පයිතන් ප්‍රකාශනයේ ඇගයීමේ නිවැරදි පටිපාටිය පහත සඳහන් කුමකින් පෙන්නුම් කරයිද?
- (1) `10-((4*10)/(2**5 * 2))` (2) `10-(((4*10)/(2**5)) * 2)`
 (3) `(10-(4*10))/(2**5 * 2)` (4) `(10-(4*10/2**5)) * 2`
 (5) `10-(((4*10)/2)**5) * 2)`

36 කාරක රීති වලට අනුකූලව නිවැරදි ව (syntactically correct) විශාලතම අගය ප්‍රතිදානය කරන පයිතන් ශ්‍රිතය වන්නේ පහත සඳහන් කුමක්ද?

- (1) `def islarger (a,b):`
`return a>b`
- (2) `del islarger (a,b):`
`return a < b`
- (3) `def islarger (a,b):`
`if a>b:`
`x = a`
`else:`
`x = b`
`return x`
- (4) `del islarger (a,b):`
`if a<b:`
`x == a`
`else:`
`x == b`
`return x`
- (5) `def islarger (a,b):`
`if a>b:`
`return = a`
`else:`
`return = b`

37 `def totalNumber (a,b,c =0,d=0):`
`total = (a + b + c + d)/4.0`
`return round(total,2)` පයිතන් ක්‍රියා පටිපාටි (Function) හා පරාමිතින් (Parameters) භාවිතය සම්බන්ධයෙන් නිවැරදි ප්‍රකාශය කුමක් වේ ද?

- A. **a** හා **b** ආවේනික තාර්කික විචල්‍යයන් (Default argument) ආකාරයකි.
- B. **c** හා **d** අනිවාර්ය තාර්කික විචල්‍යයන් (Required Argument) ආකාරයකි.
- C. `totalNumber(10,15,20)` ක්‍රියාත්මක කළ විට දෝෂ ආකාරයක් නිර්මාණය වේ.
- (1) **A** පමණක් සත්‍ය වේ.
- (2) **B** පමණක් සත්‍ය වේ.
- (3) **C** පමණක් සත්‍ය වේ.
- (4) **A, B** පමණක් සත්‍ය වේ.
- (5) **A, B** හා **C** සියල්ලම අසත්‍ය වේ.

38 `first = raw_input("Enter first number :")`
`second = input("Enter second number :")`
`print(first * second)` පයිතන් ක්‍රමලේඛය ක්‍රියාත්මක කර, අගයන් ආදානය කිරීමේ දී ඇතිවිය හැකි දෝෂ ආකාරයන් විය හැක්කේ කුමක් ද?

- A. **first** සඳහා සංඛ්‍යාත්මක අගයක් ආදානය කළ විට දෝෂ ආකාරයක් ප්‍රතිදානය කරයි.
- B. **second** සඳහා 'Ab' අගයක් ආදානය කළ විට දෝෂ ආකාරයක් ප්‍රතිදානය කරයි.
- C. පිළිවෙළින් 5 හා 'Ab' අගයයන් ආදානය කිරීම අනුව, තෙවැනි උපදේශය ක්‍රියාත්මක වීමේ දී දෝෂ ආකාරයක් ප්‍රතිදානය කරයි.
- (1) **A** පමණක් සත්‍ය වේ.
- (2) **B** පමණක් සත්‍ය වේ.
- (3) **C** පමණක් සත්‍ය වේ.
- (4) **B** හා **C** සත්‍ය වේ.
- (5) **A, B** හා **C** සියල්ලම සත්‍ය වේ.

39 විචල්‍යය හා අගයයන් භාවිතය අනුව වැරදි ප්‍රකාශය වන්නේ කුමක් ද?

- (1) `a = '20' * 2`
- (2) `number, text = false, 010`
- (3) `k1, k2 = 20, 'k1'`
- (4) `t, t = True, False`
- (5) ඉහත සියලුම ප්‍රකාශන තුළ දෝෂ පවතී.

40 `0b101 + 0101 + 0101` හි අගය නිවැරදි ව නිරූපණය වන්නේ

- (1) 171
- (2) 135
- (3) 231
- (4) 303
- (5) ඉහත කිසිවක් නිවැරදි නොවේ.

- 41 ගොනුවක (marks.py) දෙවන ඡේදයේ පවතින දත්ත කියවීම සඳහා භාවිතා කළ හැකි සුදුසුම පයිතන් ප්‍රකාශය කුමක් ද?
- | | |
|---|--|
| (1) <code>f = open('marks.txt')</code>
<code>print(f.read(2))</code> | (2) <code>f = open('marks.txt')</code>
<code>print(f.readline(2))</code> |
| (3) <code>f = open('marks.txt')</code>
<code>f.readline()</code>
<code>print(f.readline())</code> | (4) <code>f = open('marks.txt')</code>
<code>f.readline()</code>
<code>print(f.readlines())</code> |
| (5) <code>f = open('marks.txt')</code>
<code>f = f.readlines()</code>
<code>print(f[2])</code> | |
- 42 කාර්ය බද්ධ ක්‍රියා පරිපාටි (Function) නිර්මාණය හා භාවිතය අනුව දෝෂ ආකාරයන් ප්‍රතිදානය වන ක්‍රමලේඛ සම්බන්ධයෙන් නිවැරදි වන්නේ කුමක් ද?
- | | |
|--|---|
| A. <code>def average(a = 0,b,c) :</code>
<code>tmp = a+b+c</code>
<code>avg = tmp/3.0</code>
<code>return avg</code>

<code>print(average(5,10,15))</code> | B. <code>def average(a,b = 0,c =0) :</code>
<code>tmp = a+b+c</code>
<code>avg = tmp/3.0</code>
<code>return avg</code>

<code>print(average(a = 5,10,15))</code> |
| C. <code>def average(a,b,c =0) :</code>
<code>tmp = a+b+c</code>
<code>avg = tmp/3.0</code>
<code>return avg</code>

<code>print(average(a = 5,b = 10))</code> | D. <code>def average(a, b, c) :</code>
<code>tmp = a+b+c</code>
<code>avg = tmp/3.0</code>
<code>return avg</code>

<code>print(average(c = 5,b = 10, 10))</code> |
- (1) A, B පමණක් වේ. (2) B, C සත්‍ය වේ.
 (3) A,B හා C පමණක් වේ. (4) A,B හා D පමණක් වේ.
 (5) A,B,C හා D සියල්ලම.
- 43 ක්‍රමලේඛය අනුව නිවැරදි ප්‍රතිදානය කුමක් වේ ද?
- | | |
|--|---|
| <code>a , b = 'True' , 2016</code>
<code>b, c = 'ICT' , a*2</code>

<code>print("{0} {1} {0}".format(a, b, c))</code> | (1) True ICT True
(2) true ICT true
(3) True ICT TrueTrue
(4) SyntaxError
(5) ඉහත කිසිවක් සත්‍ය නොවේ. |
|--|---|
- 44 කාර්ය බද්ධ ක්‍රියා පරිපාටිය (function) තුළ ක්‍රියාත්මක වන විචල්‍යයන් සම්බන්ධයෙන් අසත්‍ය ප්‍රකාශය කුමක් වේ ද?
- (1) ස්ථානීය පරාමිතින් (Local variable) වල අගයයන් function සීමාව තුළ පමණක් ක්‍රියාත්මක වේ.
 (2) ස්ථානීය පරාමිතින් වල අගයයන් function සීමාව තුළ වෙනස් කළ හැකිය.
 (3) විශ්ව පරාමිතියක (Global Variable) අගය function සීමාව තුළ භාවිතා හෝ වෙනස් කිරීමට හැකි වේ.
 (4) ක්‍රමලේඛයක සමස්ථ සීමාව තුළ ස්ථානීය විචල්‍යයන් භාවිතා කළ නොහැකිය.
 (5) ක්‍රියා පරිපාටියක සීමාව තුළ විශ්ව පරාමිතියක් අර්ථ දැක්විය හැක.
- 45 ක්‍රමලේඛයක ඇතිවිය හැකි තාර්කික දෝෂ (Logical Error) සම්බන්ධයෙන් නිවැරදි ප්‍රකාශය කුමක් වේ ද?
- (1) ක්‍රියාත්මක කිරීමේ දී හඳුනාගත් තාර්කික දෝෂය, එම ආකාරයෙන් ප්‍රතිදානය කරයි.
 (2) ක්‍රියාත්මක කිරීමේ දී හඳුනාගන්නා ධාවන තත්වයේ දෝෂ (Run-Time Error) සියල්ල ප්‍රතිදානය කරයි.
 (3) උපදෙස් වෙනස් කිරීමකින් තොරව පවතින තාර්කික දෝෂ නිවැරදි කළ හැකිය.
 (4) වාග් රීති දෝෂ (Syntax Error) වාර්ථා වන ක්‍රමලේඛයක තාර්කික දෝෂ පැවතිය හැක.
 (5) ඉහත කිසිවක් නොවේ.

46 t1 =(5,15) Tuple දත්ත ව්‍යුහය අනුව අසත්‍ය ප්‍රකාශය වන්නේ
t2 = [1,2]

- (1) t3 = t1 + t2 ක්‍රියාත්මක කළ විට දෝෂ ආකාරයක් ප්‍රතිදානය වේ.
- (2) t3 = t1 * t2 ක්‍රියාත්මක කළ විට දෝෂ ආකාරයක් ප්‍රතිදානය වේ.
- (3) t3 = t1 - t2 ක්‍රියාත්මක කළ විට දෝෂ ආකාරයක් ප්‍රතිදානය වේ.
- (4) t3 = t1[0] + t2[0] ක්‍රියාත්මක කළ විට දෝෂ ආකාරයක් ප්‍රතිදානය වේ.
- (5) ඉහත කිසිවක් නොවේ.

47 පයිතන් print() ක්‍රියාත්මක කිරීම අනුව, නිවැරදි ව ප්‍රතිදානය කුමක් වේ ද?

print (4 == 2**2)	(1) True	(2) True	(3) False	(4) True
print (4 <> 16)	True	True	True	True
print ('4' <> 4)	True	True	True	False
print (4 == '4')	False	False	True	True
print ('4' != 4)	True	True	False	False

(5) SyntaxError හෝ Run-Time Error හෝ ප්‍රතිදානය වේ.

48 ict = 'Python is a my word' අනුව, නිවැරදි ව ක්‍රියාත්මක නොවන පයිතන් උපදේශය

- (1) ict = ict[0:5][0:2]
- (2) ict += ict[0:5][2]
- (3) ict = ict[-4:][:]
- (4) ict = ict[0:8][:6]
- (5) ඉහත කිසිවක් නොවේ.

49 පයිතන් loop භාවිතය සම්බන්ධව අසත්‍ය ප්‍රකාශය කුමක් ද?

- (1) දන්නා වාර ගණනක් උපදෙස් ක්‍රියාත්මක කිරීම සඳහා for loop භාවිතා කරයි.
- (2) while loop භාවිතා කර තිබෙන ක්‍රමලේඛයක් එම ප්‍රතිදානය ම ලැබෙන පරිදි for loop මගින් සකස් කළ හැකිය.
- (3) ඇතම් while loop ප්‍රකාශන, for loop භාවිතයෙන් එම ප්‍රතිදානය ම ලැබෙන පරිදි සකස් කළ නොහැක.
- (4) දන්නා වාර ගණනක් උපදෙස් ක්‍රියාත්මක කිරීම සඳහා while loop භාවිතා කළ නොහැක.
- (5) for loop භාවිතයෙන් අපිරිමින පුනරාවර්තන (infinite Loop) සඳහා උපදෙස් සකස් කළ නොහැක.

50 ක්‍රමලේඛය ක්‍රියාත්මක කිරීම අනුව ලැබිය හැකි ප්‍රතිදානය කුමක් වේ ද?

```
for i in range(1, 7):
    for j in range(6, 0, -1):
        if j <= i: print(i), ;
    print('')
```

- | | | | |
|-----|--|-----|--|
| (1) | 1
2 2
3 3 3
4 4 4 4
5 5 5 5 5 | (2) | 1
2 1
3 2 1
4 3 2 1
5 4 3 2 1 |
| (3) | 1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
6 6 6 6 6 6 | (4) | 1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
6 5 4 3 2 1 |
| (5) | 2
3 3
4 4 4
5 5 5 5
6 6 6 6 6 | | |